SYSTEMS ENGINEERING RESEARCH CENTER

RT 164: Design and Development Tools for the Systems Engineering Experience Accelerator – Part 3

Technical Report SERC-2017-TR-107

April 29, 2017

Principal Investigator: Dr. Jon Wade, Stevens Institute of Technology

Co-Principal Investigator: Dr. Doug Bodner, Georgia Institute of Technology

Research Team:

Defense Acquisition University: Yvette Rodriguez

Georgia Institute of Technology: Jing Liu

Stevens Institute of Technology: Dr. Richard Turner

Stevens Institute of Technology: Peizhu Zhang

Sponsor: Office of the DASD (Systems Engineering)

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract H98230-08-D-0171 and HQ0034-13-D-0004 (TO 0064).

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

No Warranty.
This Stevens Institute of Technology and Systems Engineering Research Center Material is furnished on an "as-is" basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The Department of Defense faces many challenges related to deploying new systems to meet the needs of the warfighter. Among these are compressed development times and increased capabilities required. There is a real need for systems engineering expertise in the acquisition enterprise to guide the technical design, development, integration and testing of such systems.

At the same time, there is a potential skills gap due to an aging workforce, many of whom are near retirement, and the experience curve needed to bring new talent up-to-speed. Often, this experience curve takes many years and spans multiple programs. The idea behind the Systems Engineering Experience Accelerator is to accelerate the experience curve so that new talent can become proficient much more quickly to meet DoD needs. This is done by using educational technology, and specifically the notion of role-playing in immersive environments where a learner not only leans the technical skills associated with a systems engineering position, but also critical skills in persuasion and decision-making needed to deploy technical skills effectively.

The original SEEA was developed to let the learner play the role of a chief systems engineer in an acquisition program for a new unmanned aerial vehicle system. It was developed using standard programming tools. Clearly, though, there are many different systems engineering skills and competencies needed beyond one such educational experience. The goal then became t develop a library of such experiences. To do so quickly and cost efficiently, it was determined that a higher-level set of tools were need to support development, especially tailored to individuals without substantial programming expertise. Thus, an effort was initiated to create a suite of such tools.

This report details the third part of development of these tools. They consist of three suites of tools. Simulation Modeler enables an experience designer create and tune simulation models and output charts that advance the state of a simulated world with which the learner interacts in the role of a systems engineer. Experience Builder lets the designer specify the different learning phase, cycles, events and communications experienced by the learner. Learning Assessor lets the designer determine what learning data to capture and how to analyze it to determine the effectiveness of different learning experiences. In addition, this report describes an upgrade of the Experience Accelerator infrastructure to HTML5, which allows for more flexibility in developing functionality and interfaces for different experiences, and is critical for supporting compliance with regulations governing accessibility (Section 508 compliance). The report describes evaluation of the tools by active users.

The risk and mitigation plan worked well in supporting the successful completion of all the objectives of this task that did not have dependencies on other work. In addition, a new capability not envisioned at the beginning of this project, the NPC Editor, was developed and evaluated. The report concludes with recommendations for future work.

INTRODUCTION

Educational technology is used increasingly in a variety of domains, ranging from high school and college-level courses, to on-the-job training, to advanced skills development. One of the appealing features of education technology is that it can be used not only to convey the material being taught, but also the context in which the material should be applied. This can be accomplished through role-playing environments in which a learner interacts with an automated system providing the context, or it can be accomplished through technologies that connect multiple learners who interact to solve problems and apply solutions in context. Systems engineering is one field for which educational technology has potential for application.

CHALLENGES IN SYSTEMS ENGINEERING EDUCATION AND WORKFORCE DEVELOPMENT

Systems engineering is a multidisciplinary practice and is much of an art as it is a science. While a waterfall model of education can provide a background of domain-centric knowledge, it is not until this knowledge is put into practice in an integrated, real world environment that a Systems Engineering can develop the necessary insights and wisdom to become proficient. In the workplace, these learning events are often distributed sparsely over time such that an engineer may only see a complete system life cycle over a period of several years. As a result, the maturation time from completion of formal studies to becoming seasoned Systems Engineer is unacceptably long, particularly when contrasted with the clock speeds of today's society in which career change is the norm rather than the exception, particularly among the young.

Clearly, there is a critical need to promote rapid skill development of systems engineers, in particular senior systems engineers, across the DoD and government workforces, as a large cohort of personnel is nearing retirement age. In addition, new systems engineering skills are needed to address important societal needs in national security, homeland security, airspace management and disaster recovery. These domains involve large-scale, systems-oriented solutions with increasingly limited budgets. Educational technologies hold the promise of providing customized learning exercises based on real-world situations to reduce the reliance on extensive on-the-job training that is the hallmark of current workforce development.

EXPERIENTIAL LEARNING AND THE SYSTEMS ENGINEERING EXPERIENCE ACCELERATOR

Prior work resulted in the Systems Engineering Experience Accelerator (SEEA), a technology platform that supports experiential learning for systems engineers. This technology platform created a new approach to developing the systems engineering workforce that augments traditional, in-class education methods with educational technologies aimed at accelerating skills and experience with immersive simulated learning situations that engage learners with problems to be solved. Although educational technology is used in a variety of domains to support learning, the SEEA is one of the few such technologies that supports development of the systems engineering workforce.

The SEEA was developed to support a single-person role-playing experience in a digital environment, as well as a specific learning exercise in which a learner plays the role of a lead systems engineer for a Department of Defense (DoD) program developing a new unmanned aerial system. This exercise is based on the notion of experiential learning, and thus will be referenced as an experiential learning module. The learner engages with the experience (i.e., simulated world), makes decisions to solve problems, sees the results of those decisions, abstracts lessons learned from what was successful and what was unsuccessful, and then repeats the process in a series of cycles, simulating the evolution of the program over time.

The SEEA technology provides a graphical user interface allowing the learner to see the program status, interact with non-player characters to gain additional program information, and make technical decisions to correct problems. It also provides the capability to simulate the program into the future, based on these learner decisions, so that outcomes can be shown to the learner. This cycle of decision and simulation-into-the-future supports the Kolb cycle of experiential learning; the Experience Accelerator uses multiple such cycles operating through the lifecycle of the program. In particular, this approach allows illustration of the effect of upstream decisions on downstream outcomes in the system lifecycle. The SEEA can support a wide variety of systems domains and areas of expertise through changes to the experience. Recently, additional multi-player technology is being developed to allow live player support for team-based learning, as well as for a mentor to provide advice and feedback.

Development Process for Learning Experiences

The first UAV experience was developed in tandem with the SEEA technology platform. Since it was a first-time effort, there were no tools other than programming environments to support its development. This development was a time-consuming process.

The eventual goal is to have a variety of experiences supported through the SEEA technology to meet the needs of many learners. These experiences would focus on different types of systems – naval vessels, satellite systems, air traffic control, etc. In addition, experiences would focus on different phases of the system lifecycle, from requirements development, to concept selection, to development, to production and sustainment. Finally, the SEEA technology is being made available to a variety of organizations for use in their specific educational and professional development programs. Thus, it made sense to develop a suite of tools to aid experience developers in creating new experiences or modifying existing ones.

This report addresses the third part of an effort that is creating a tool suite for experience developers (Wade et al., 2016). In the early part of the overall effort, the tools were identified in relation to the experience components used in the SEEA. The tool suite consist of the following:

- Simulation Modeler – GUI tool for building and testing system dynamics simulation models
    - Sim Builder – Simulation model builder using libraries/templates
    - Sim Tuner – Parameter tuner that automates the tuning of parameters to yield desired out-puts via batch processing of different combinations of settings

  - Chart Designer – Automates design of simulation output charts
- Experience Builder – Integrates the Phase and Event Editor, and the Artifact Integrator
  - Phase Editor – GUI-based tool for phase, cycle and event specification, with code generation
  - Event Editor – GUI or text-based tool to specify events and their triggers, with code generation
  - User Interface Editor – GUI tool to support tailoring the user interface to the specific experience and simulation(s) involved
  - Artifact Integrator – Application that allows designer to take artifact files, such as design documents and enter them into the EA application with automatic recompilation and re-linking
- Learning Assessor – Assessment tool-suite that provides automated performance scoring and decision comparisons against proven baselines

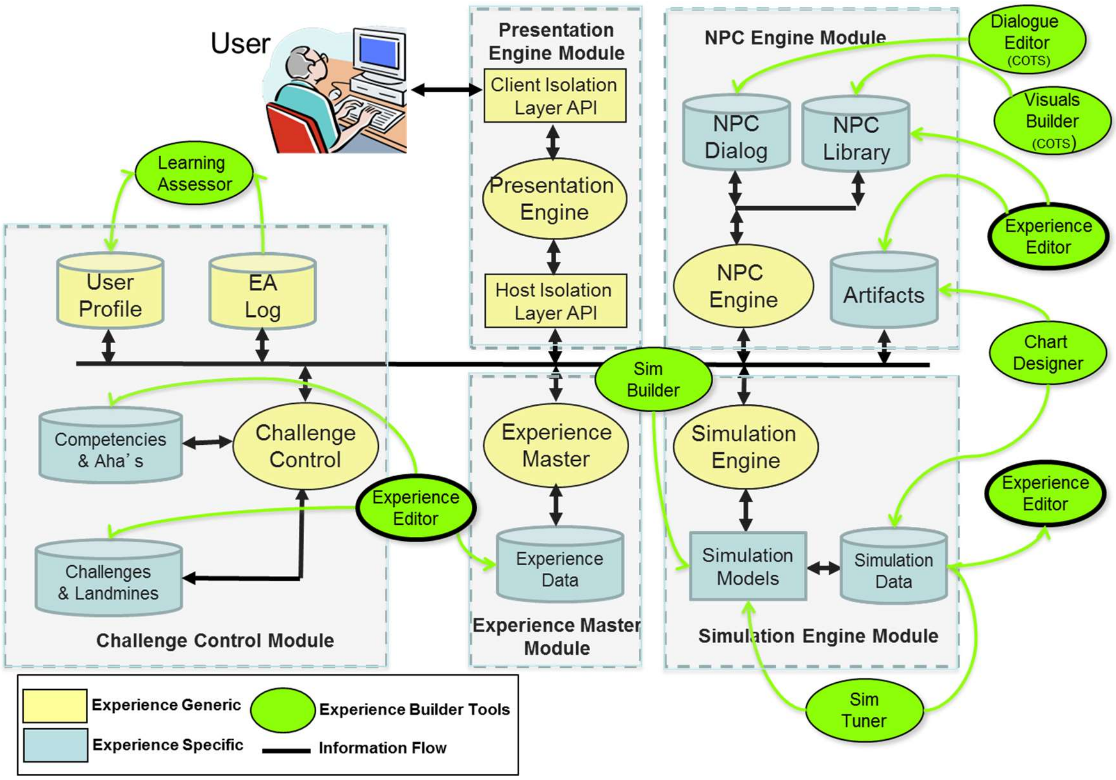


Figure 1. Relationship between tools and SEEA components

Each of the tools maps to a particular set of components in the overall SEEA architecture, as shown in **Error! Reference source not found.**. For instance, the simulation tools map to the simulation models and chart outputs (artifacts). The experience builder tools map to experience master, NPC library and dialog, and artifacts.

In addition, the SEEA learner interface technology needs to be upgraded from Flash to HTML5. This is critical for Section 508 compliance, and is a requirement for U.S. Government use.

Finally, the tools will be evaluated by users who intend to develop their own experiences apart from the current UAV experience.

The remainder of this report is organized as follows. The research hypothesis is discussed first, along with measurable outcomes and the overall research goal. Then each set of tools is presented in turn – Simulation Modeler, Experience Builder, and Learning Assessor. Afterward, the upgrade of the SEEA infrastructure to HTML5 is described. The evaluation of the tools is provided next. Finally, the Conclusions and future research are presented.

RESEARCH OVERVIEW

This project seeks to create a set of tools to enhance capabilities to rapidly and cost-effectively create experience modules that can be used by the current generic Experience Accelerator technology set to deliver a variety of learning experiences tuned to the needs of particular workforces. In addition to evolving the tools that were developed in EA Tools Parts 1 and 2, the EA infrastructure needs to be updated by having the EA user interface transitioned from Flash to HTML5. This transition will provide the ability to use a wide variety of HTML5 development tools, greatly improving the ease of rapidly updating the learner interface and its artifacts. HTML5 also facilitates the development of artifacts and user interface updates, enables the SEEA to support a wider range of client devices (iPads, iPhones, etc.) and is critical for Section 508 compliance, which is a requirement for US Government use.

Problem Statement: Traditional systems engineering education is not adequate to meet the emerging challenges faced by the Department of Defense and others in system acquisition and sustainment. New educational technologies such as the Systems Engineering Experience Accelerator hold the promise of facilitating a rapid skill and experience accumulation for the workforce to meet these challenges. However, to have scalable effect, such technologies cannot rely on extensive programming and low-level code development to create a rich set of experiential learning modules needed to accelerate systems engineering workforce development.

Hypothesis: The Experience Accelerator technology will scale to support a community of developers engaged in creating modules for their organizations' use if tools are developed that allow educators and other non-programmers to create, maintain and evolve experiential learning modules.

Measurable Outcomes: The outcomes from this research will be measured in two main ways. First, educators and others interested in creating experiential learning modules will provide qualitative feedback on the effectiveness and efficiency of the Experience Accelerator toolset in creating experiential learning modules based on their use of the design and development tools. Second, the number of such developers who commit to create modules for their organizations' use will be tracked, as well as the number of organizations and variety of different application areas.

Research Goal: Validate the hypothesis through the creation of design and development tools for experiential learning modules that maximally leverage the current Experience Accelerator research, technology and content.

The following is a description of the features and capabilities that will be developed in Part 3. All of these feature have been completed with the exception of the development of additional phases of simulation library sub-models, and the release of the tools. The Experience Builder NPC Editor was developed, which was not on the original development list in the proposal. These three items are noted below.

Simulation Tools

Sim Builder

- Initial library of sub-models. An initial library of sub-models for experiences will be created.
    - The simulation model that supports the Phase-2 portion of the current UAV experience will be re-designed so that selected elements are modeled via sub-models. These sub-models will be archived into an initial library.
    - Additional phases of the current experience will likewise be developed in a modular manner using sub-models, and these will be archived. – *This work is dependent on RT167 and will be completed when the dependencies are completed.*
    - A set of generic sub-models for systems engineering modeling will be identified and developed for the library.

Sim Tuner

- Intelligence for variable/constant changes. The user should receive intelligence as to the effect of changes in various variables and constants on other variables based on dependencies.
- Obviousness of changes. When the user makes a change, the effect should be made obvious, for instance, using a before-and-after representation.

Chart Designer

- Interface for chart design. The user should have an intuitive interface for designing charts, building on the initial prototype.
    - Provide means for specifying scaling of axes (e.g., zero-to-max, min-to-max, min-to-max-plus-10%-on-each-side, etc.).
    - Provide means for specifying program vs. phase durations.
    - Provide means for specifying plans on charts for TPMs/KPPs, etc.

Release

- Transition tools. Transition simulation tools to the INCOSE system dynamics community. *– This will be done after the completion of project.*

Experience Builder Tools:

- Documentation/guidance for end to end experience building using the toolset
- HTML5 tool identification and pilot use for building interactive UIs (e.g., screens, dashboards, recommendation forms, etc.)
- Non-Player Character Editor for creating and editing NPCs. *This was an addition to the original work plan.*

Learning Assessor Tool:

- New user interface for Learning Assessor Tools

EA Infrastructure:

- Conversion from Flash to HTML5
  - Move appropriate client functionality to server
  - Conversion of current artifacts
  - Rebuild dialog engine for HTML5

SIMULATION MODELER

The simulation modeler provides tools for the experience designer to create simulation models used in the SEEA. These simulation models are executed by the SEEA to advance the state of the simulated world in time. For instance, in the UAV experience, the simulated world consists of the development program. The simulation models represent this world and advance its state over time through execution one of the models.

The learner has control over some of the variables in the simulation model. At decision points during the experience, the learner enters recommendations, which may change the values of variables or parameters in the simulation. Such changes alter the state trajectory of the simulated world. Ideally, such changes would address problems. However, the learner may discover that the problems are not addressed adequately, or that new problems are created.

The simulation models are based on the systems dynamics paradigm of simulation (CITE). In system dynamics, a flow system is used to model state changes in the world. Rates at which flow occurs are governed by equations. These equations can be used to model positive feedback loops, in which a particular phenomenon may grow. Likewise, they may be used to model negative feedback loops, in which a steady-state is achieved. In addition to feedback loops, such phenomena as lags and communication overhead can be represented.

System dynamics has been applied a variety of domains. Relevant to the SEEA are domains such as project management and earned value management (CITE).

Prior Work

Previously, two main tools were developed under the Simulation Modeler (CITE prior reports). The Sim Builder provides a graphical user interface for building systems dynamics models. Prior work focused on improving the existing tool by adding sub-models and other features to enable model modularity and reuse through model component libraries. The Sim Tuner allows the experience designer to execute simulation models outside of the SEEA to see how they behave over time. Variables and parameters can then be adjusted to achieve desired behavior. In addition, the designer can take the role of the learner and test different possible learner recommendations to see how they impact the overall performance measures of interest in the experience.

In addition to the Sim Builder and Sim Tuner, a prototype tool was developed to allow the experience designer to specify charts. This tool, the Chart Designer, was matured in the current effort.

Sim Builder

Work on the Sim Builder focused on developing sub-models, with the following tasks being identified.

1. Initial library of sub-models. An initial library of sub-models for experiences will be created.

A variety of sub-models have been created for the sub-model library. These focus on the Phase-2 simulation model for the current UAV experience and include technical performance and key performance parameter (TPM/KPP) sub-models and earned value management (EVM) sub-models. Figure 2 shows two sub-models. The one on the left is the range sub-model for the UAV development effort. The one on the right is the weight sub-model. The weight sub-model addresses weight growth and weight reduction efforts for the empty UAV. It feeds into the range sub-model via a shared node (shown in green).

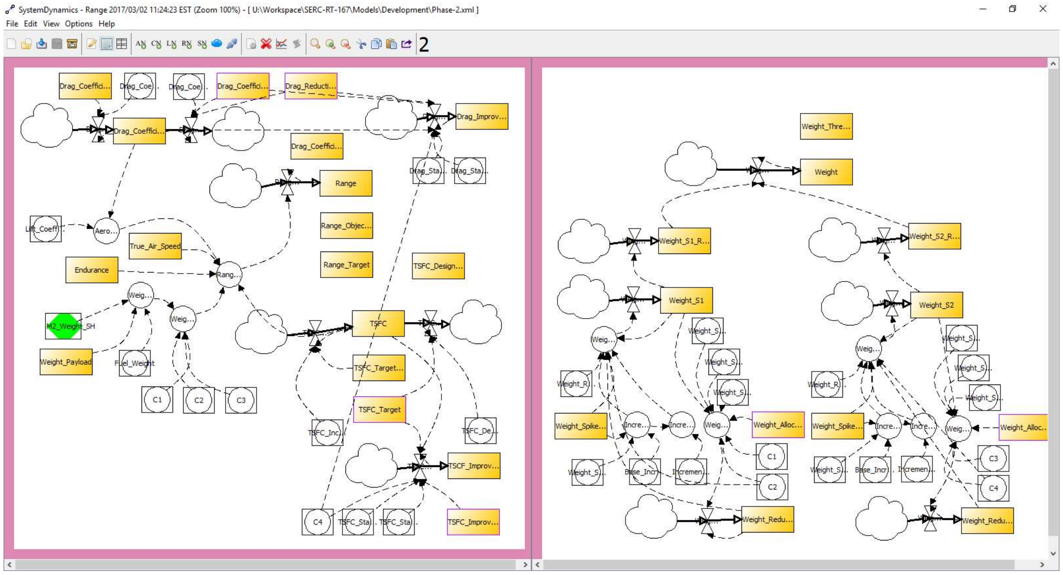


Figure 2. TPM/KPP sub-models

In addition to the sub-model development, two code development tasks were identified and addressed. **These are in addition to the original work plan.**

- The interface for the Sim Builder was modified so that the user can select a sub-model and open a new window with just that sub-model for editing. This is a useful feature for editing complex sub-models.

- A new feature was added to the variable initialization between phases allowing more complex functions to be included.

Sim Tuner

New functionalities for the Sim Tuner focused on aiding the experience designer when he or she is tuning a model. The following tasks were identified.

1. Intelligence for variable/constant changes. The user should receive intelligence as to the effect of changes in various variables and constants on other variables based on dependencies.

2. Obviousness of changes. When the user makes a change, the effect should be made obvious, for instance, using a before-and-after representation.

A new feature was implemented to provide the experience designer with intelligence on the relationships between variables. Given an even moderately complex model, such relationships are not necessarily obvious. The Sim Tuner executes the simulation and graphs variables of interest (selected by the experience designer) for testing a model. When the graph is displayed, the interface also displays a legend above the graph denoting first-order relationships between the variable of interest and other variables. This is depicted in Figure 3. The variable ACWP_S1 is being graphed. It is shown as being dependent on parameters Labor_Cost_Rookies_S1 and Labor_Cost_Pros_S1. This is a positive relationship, so that if these two parameters are increased, then ACWP_S1 increases.

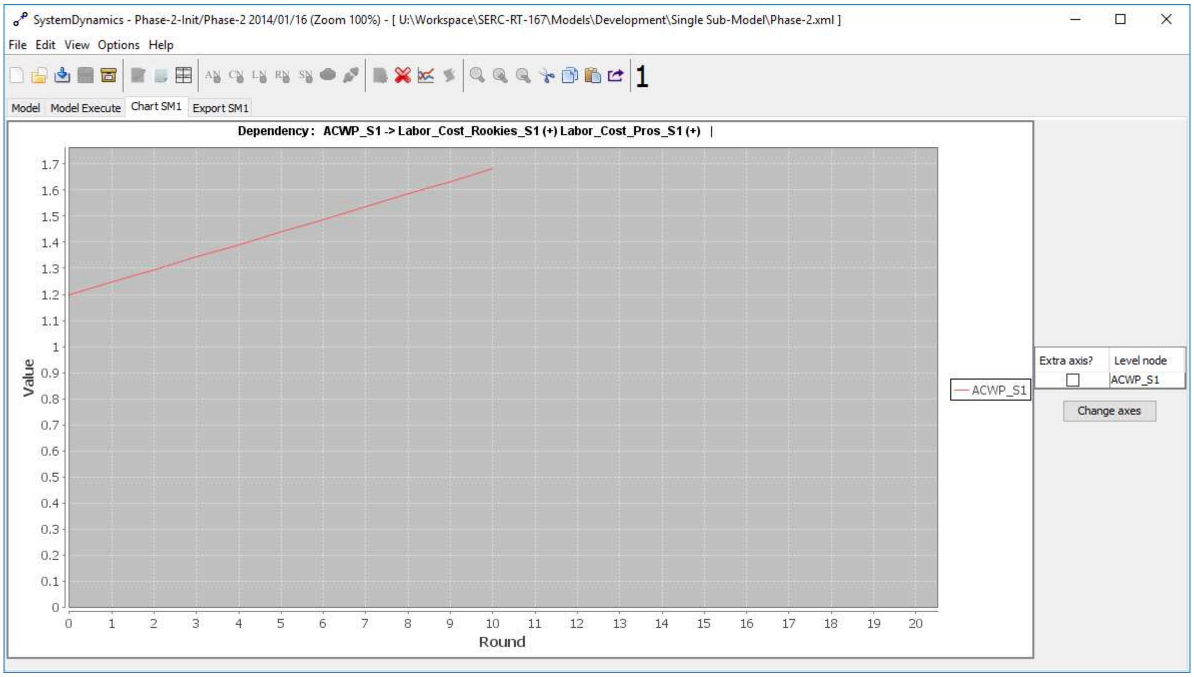

SystemDynamics - Phase-2-Init/Phase-2 2014/01/16 (Zoom 100%) - [U:\Workspace\SERC-RT-167\Models\Development\Single Sub-Model\Phase-2.xml]
File
Edit
View
Options
Help
1
Model
Model Execute
Chart SM1
Export SM1
Dependency : ACWP_S1 -> Labor_Cost_Rookies_S1 (+) Labor_Cost_Pros_S1 (+) |
Value
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Round
ACWP_S1
Extra axis?
Level node
☐
ACWP_S1
Change axes

Figure 3. Intelligence for variable relationships

This feature works only for first-order relationships due to the complexity of the logic that would be required to represent the potential higher order relationships (e.g., variable A depends on variable B, which depends on variable C, etc.).

To support the obviousness of changes, a new interface system was developed. When examining how a change in a variable (or variables) impacts model output, the experience designer executes the model with an initial set of values for these variables in the Sim Tuner. The designer can then return to the Sim Builder interface to enter a new set of values. After performing this function, the designer then returns to the Sim Tuner interface to execute the model with the new values. The output chart interface contains two charts, one for the initial variable values and one for the revised values.

Figure 4 shows the new interface for variable change analysis. The lower chart shows the original output. The upper chart shows output using revised values of variables.

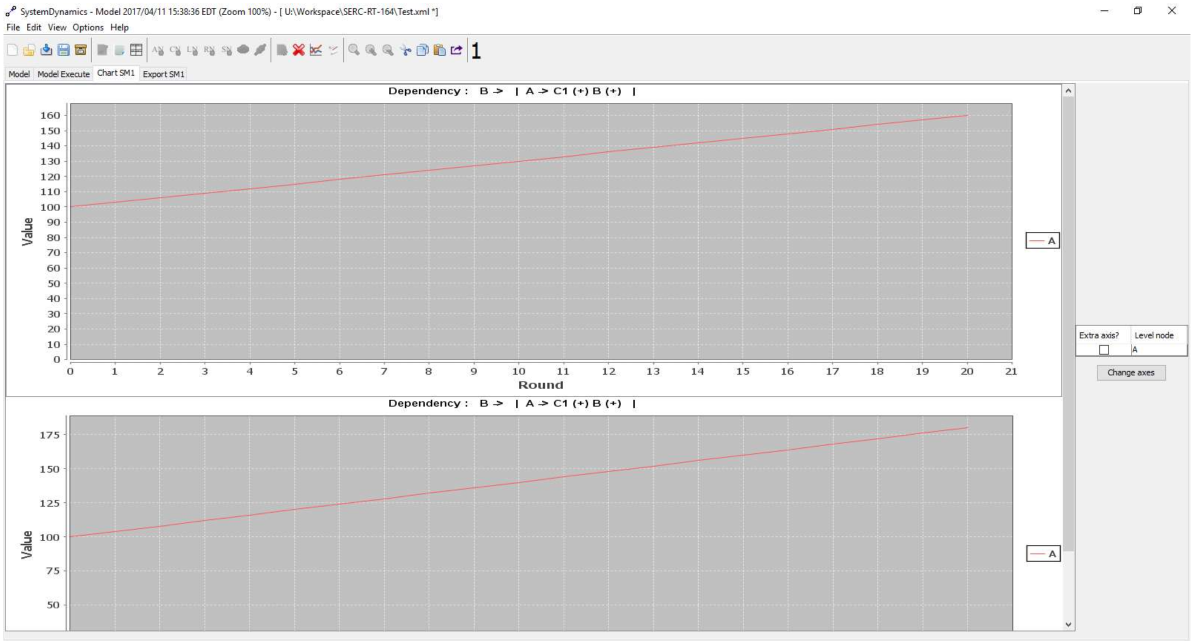

SystemDynamics - Model 2017/04/11 15:38:36 EDT (Zoom 100%) - [U:\Workspace\SERC-RT-164\Test.xml *]
File Edit View Options Help
1
Model
Model Execute
Chart SM1
Export SM1
Dependency : B -> | A -> C1 (+) B (+) |
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
Value
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Round
A
Extra axis?
Level node
A
Change axes
Dependency : B -> | A -> C1 (+) B (+) |
175
150
125
100
75
50
Value
A

Figure 4. Interface for variable change analysis

CHART DESIGNER

The main goal of this effort with respect to the Chart Designer was to mature the prototype developed previously. The following tasks were identified.

1. Interface for chart design. The user should have an intuitive interface for designing charts, building on the initial prototype.

    - Provide means for specifying scaling of axes (e.g., zero-to-max, min-to-max, plus-10%-on-each-side, etc.).
    - Provide means for specifying program vs. phase durations.
    - Provide means for specifying plans on charts for TPMs/KPPs, etc.

The initial graphical user interface was completely redesigned, since the original one was implemented mainly for demonstrating proof-of-concept. Since the chart specification files use an XML schema, the graphical user interface was designed to support creating and editing the features above.

Figure 5 shows the interface for the Chart Designer. The experience designer selects the Chart Designer from the main Sim Builder interface, it creates a new chart file associated with the model file that currently is open. The designer can then start populating different charts. Each chart has a set of information associated with it that is entered first – name, chartID, file name (for output file), x-axis label, y-axis label, global (i.e., whether the chart is for the current experience phase or whether it spans the different phases), and pr (i.e., whether the chart has a variable that will be used as the 100% mark so that other variables can be compared to it using a percentage scale on the right-side y-axis).

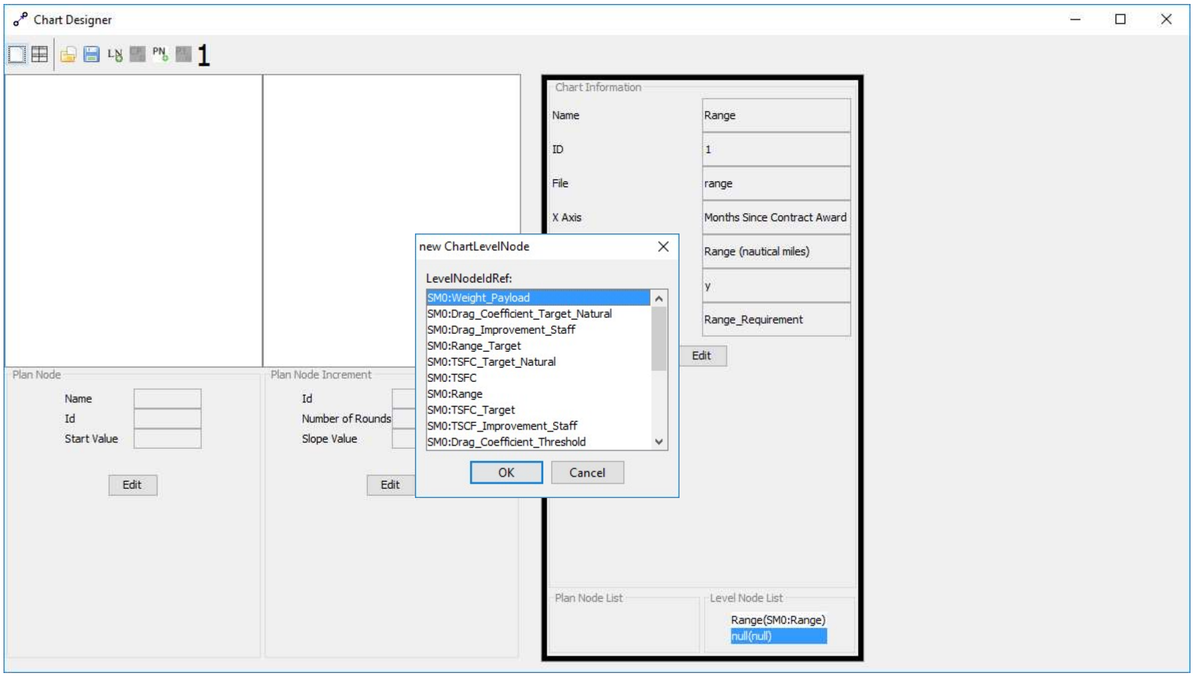


Chart Information	
Name	Range
ID	1
File	range
X Axis	Months Since Contract Award
	Range (nautical miles)
	y
	Range_Requirement

LevelNodeIdRef:

SM0:Weight_Payload
SM0:Drag_Coefficient_Target_Natural
SM0:Drag_Improvement_Staff
SM0:Range_Target
SM0:TSFC_Target_Natural
SM0:TSFC
SM0:Range
SM0:TSFC_Target
SM0:TSCF_Improvement_Staff
SM0:Drag_Coefficient_Threshold

Level Node List:

Range(SM0:Range)
null(null)

Figure 5. Chart designer interface

In Figure 5 above, the chart information has been entered. The pop-up dialog is now allowing the designer to select variables from the model file to be graphed in the chart. The designer will be queried to provide a label for the variable that will be used in the chart legend.

The designer can also enter plans for the different variables being charted. For instance, this may correspond to a plan for the weight of an aircraft system during development. There may be some anticipated weight growth that needs to be included. Since more than one chart may share a particular plan, the Chart Designer lets the experience designer create a library of plans. This library is on the left side of the interface. Each plan consists of a series of segments, each with a length and slope. Thus, the interface accommodates the designer's entering a series of connected line segments. The designer can then select a plan from this library for inclusion in any particular chart.

Figure 6 illustrates one of the simulation output charts. This chart shows the weight of an aircraft system as the program unfolds. The current design weight is being tracked. The weight threshold maximum is the upper limit on allowable values for the weight. It is pegged at 100% on the right-side y-axis (i.e., it is set as the "pr" variable for the chart). Thus, one can see how the current design weight relates to this threshold as a percentage instead of only being able to compare their magnitudes. The weight plan is allowing for weight growth. However, the current weight is spiking well above the plan, indicating an issue to be addressed. This particular chart covers the various phases for the program (i.e., the "global" indicator is set to "y" for this chart). It could also be set so that just charts the particular phase (from PDR to CDR).

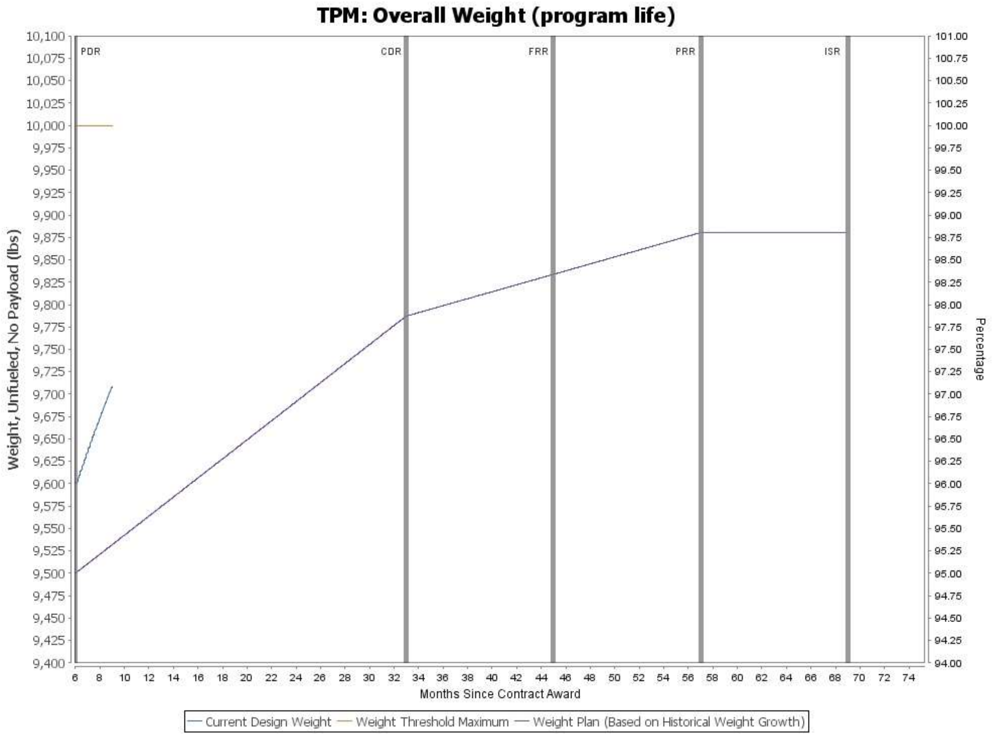


Figure 6. Example chart

Once the experience designer is finished specifying a chart, the chart file is saved and is available for use in the SEEA.

Unit Testing and Integration Testing

During the development of the Simulation Modeler tools, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

Experience Building Tools

The experience building tools provide multiple options for experience designers to alter different aspects of the experience. The tools utilize the SEEA architecture design and allow experience designers to change the built-in experience. They also provide the capabilities necessary to create a new experience from scratch.

The target users of the experience building tools are educators and experience designers without significant knowledge of the SEEA design. There is no requirement for the target users to have programming skills. Therefore, the experience building tools utilize a user friendly design with a graphical user interface and simple actions such as drag and drop.

The experience building tools including the Phase Editor, Event Editor and Artifact Integrator are focusing on the implementation of the experience design. The Phase Editor allows experience designers to customize the experience flow. The Event Editor provides experience designers with the ability to create scripted events. Finally, the Artifact Integrator supports the integration of new materials into an experience with permission control and content creation functionalities.

PRIOR WORK

Previously, three tools were developed under the Experience Building Tools. The Phase Editor provides a graphical user interface for building the SEEA experience from the finite state machine perspective. Prior work focused on completing the main features of the existing tools. The Event Editor allows the experience designer to create, edit and manage experience events that can be used by the SEEA. During this research period, the Event Editor was refined. The Artifact Integrator tool provides the experience designer with a graphical means to manage artifacts in an experience; user files, emails and voicemails can be managed and adjusted to achieve desired experience.

All of the tools have matured during this research period. New features were added to the integrated tool set, which were used, evaluated and refined.

INTEGRATED TOOLSET

During this research period, the three major tools were combined into a single integrated toolset. The major functionalities were combined and integrated. This effort reduced abundant code and provided a streamlined user experience.

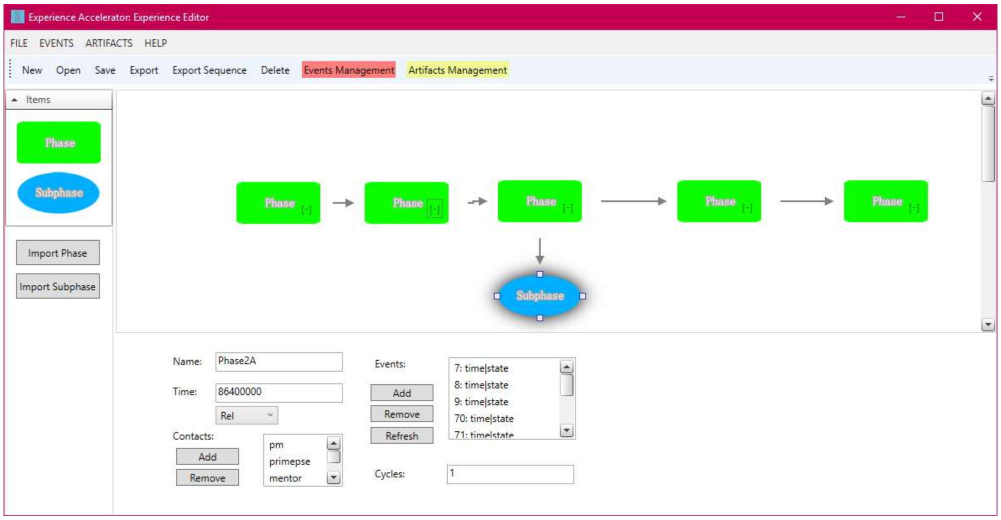
The experience building tools provide multiple options for experience designers to alter different aspects of the experience. The tools utilize the SEEA architecture design and allow experience designers to change the built-in experience. They also provide the capability to create a new experience from scratch.

The target user of the experience building tools are educators and experience designers without significant knowledge of the SEEA design. There is no requirement for them to have programming skills. Therefore, the experience building tools utilized user friendly design with graphical user interface and simple actions like drag and drop.

Phase Editor – This tool provides the ability to change the finite state machine that controls the phases within an SEEA experience. For example, the project phases can be customized to new domains and environments and can be constructed to represent state changes that are not affiliated with formal project states. For existing experience modification, the Phase Editor can be used to change the available events, available NPCs and number of cycles for a specific phase. Figure 7 shows the graphical user interface of the Phase Editor.

The Phase Editor presents the experience designer with a canvas that utilizes drag and drop action to easily create experience flows with phase and sub-phases. For each phase and sub-phase, an experience designer can specify name, starting time, available NPCs, events that may be triggered and the number of cycles learner will go through. By connecting phases and sub-phases on the canvas, an experience designer could create a phase sequence for an experience that can be saved and exported for later use.

The Phase Editor also supports modification of existing phase/sub-phases. Importing an XML formatted Phase Data File will input the configuration of an existing phase with name, time, NPC and events data for modification.

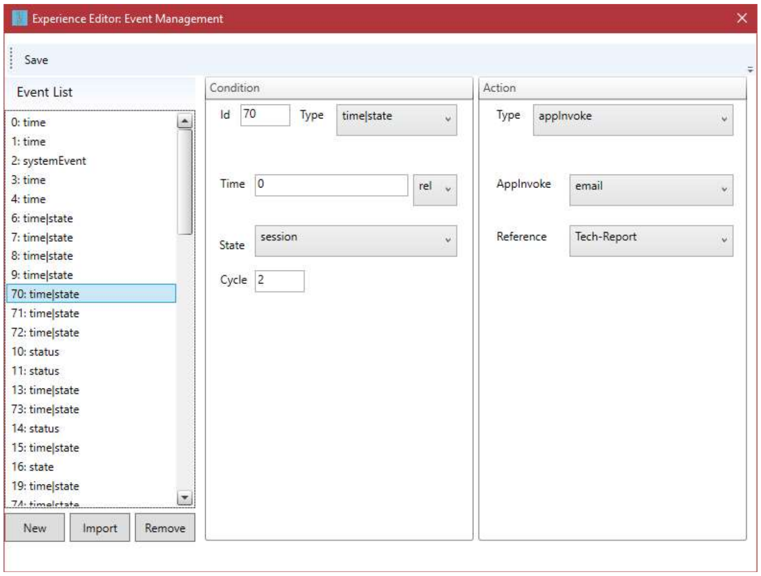


Name: Phase2A
Time: 86400000
Rel
Contacts: pm, primepse, mentor
Events: 7: time|state, 8: time|state, 9: time|state, 70: time|state, 71: time|state
Cycles: 1

Figure 7: Phase Editor User Interface

Event Editor – This tool provides the capability to create and edit events during an experience and the activities that may trigger them. For example, a phone call from the learner's supervisor can be triggered based on a decision made by the learner or the state of the project. Figure 8 shows a screenshot of the Event Editor.

By using the Event Editor, an experience designer can create/modify events using the condition and action panel. Inside the condition panel, an event ID, condition type and properties related to the selected event type are needed. Condition type indicates the type of the triggers that will be fired. The action panel provides the options of action type and action properties.



0: time
1: time
2: systemEvent
3: time
4: time
6: time|state
7: time|state
8: time|state
9: time|state
70: time|state
71: time|state
72: time|state
10: status
11: status
13: time|state
73: time|state
14: status
15: time|state
16: state
19: time|state
74: time|state

Id 70 Type time|state

Time 0 rel

State session

Cycle 2

Type appInvoke

AppInvoke email

Reference Tech-Report

Figure 8: Event Editor User Interface

Artifact Integrator – This tool provides an experience designer with the ability to quickly upload an experience change, be it a new artifact such as a document, report, or a change phase and or event, and test the results without having to do any programming. Figure 9 shows the graphic user interface of the Artifact Integrator.

The functions of Artifact Integrator are separated into four groups as follows:

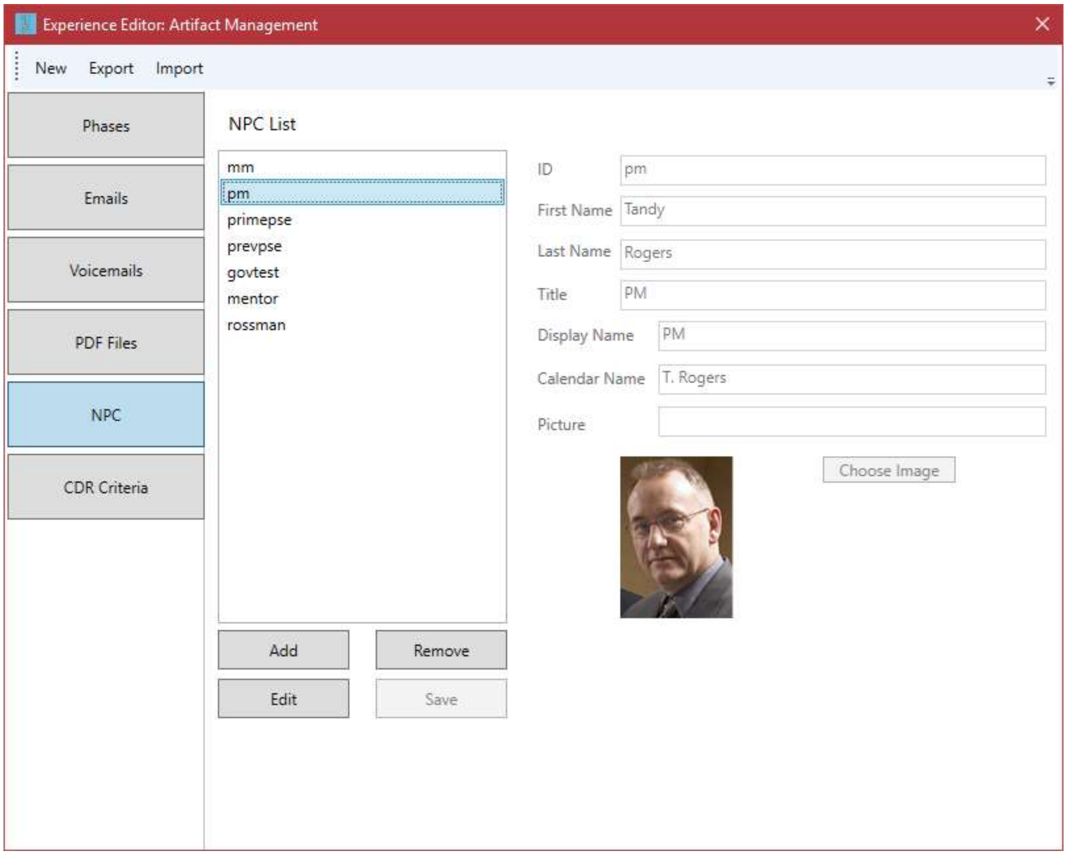
- **Learner File System by Phase** - This function provides an experience designer with the ability to set the availabilities of textual and graphical materials by phase. It is also possible to create permission settings that grants learners access to specific materials only when certain criteria are met.

- **Emails** - The Artifact Integrator provides functions to create, edit and remove emails in a SEEA experience. Experience designers can create new emails by adding email ID, sender/contact, subject and email body. The Email body supports the use of variables that support the creation of dynamic content based on experience status.

- **Voicemails** - Voicemail creation is similar to that of the email creation. However, voicemails can only be sent by a Non-Player Character (NPC), whereas emails can be sent by other learners as well.

- **PDF File Conversion and Integration** - Since the technology used in the client-side graphical user interface does not directly support PDF formatted files, the Artifact Integrator supports an embedded PDF file conversion function to reduce the effort necessary to add new learner accessible materials to the experience.

- **NPC Character** - This function enables the experience designer to create new non-player-characters. Available character properties are ID, first name, last name, title, display name, calendar name, and picture.



mm
pm
primepse
prevpse
govtest
mentor
rossman

ID: pm
First Name: Tandy
Last Name: Rogers
Title: PM
Display Name: PM
Calendar Name: T. Rogers
Picture:

Figure 9: Artifact Integrator User Interface

UNIT TESTING AND INTEGRATION TESTING

During the development of the Experience Building tools, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

LEARNING ASSESSMENT TOOLS

The Learning Assessment Tools measure the efficacy of the experiences by analyzing the data recorded throughout the learners' participation. Traditionally, learning assessment has been done through examinations and experts' reviews and opinions on students' work. However, most approaches emphasize comparing learners' performance against those of the experts' and less about the evaluating the actual learning performance of individuals. There has been much research in the domain of systems engineering education attempting to find the best way to assess students' understanding and learning about systems engineering [17-30]. Though simulation has been widely adapted by systems engineering learning, it has yet to be used to assess learner competencies and learnings performance in systems engineering and technical leadership learning.

Learning assessment is a critical component of accelerated learning. It is imperative to understand individual learning and the efficacy of the various learning experiences. This is critical both in determining the capabilities of the learner, but also enable the continual improvement of the capabilities of the learning experience.

PRIOR WORK

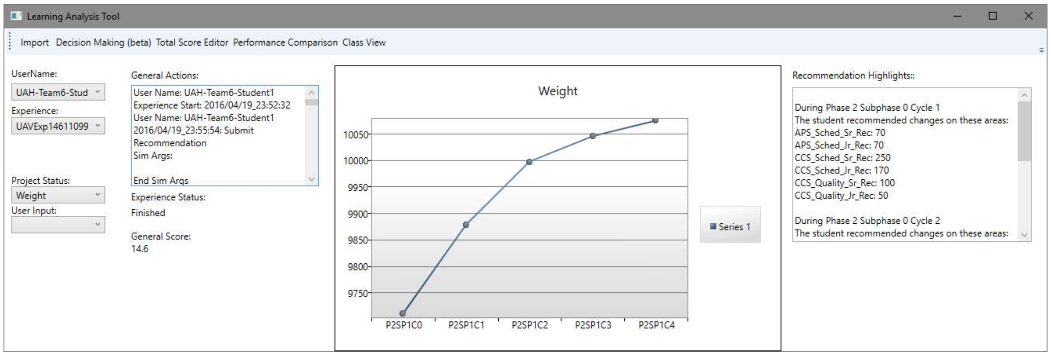
In previous phases, research was done to create the high-level design of the Learning Assessor. A Learning Assessor prototype was then developed and demonstrated. The current effort involves the continued development of the Learning Assessor tool, assessing its capabilities through external evaluation, and determining desired features for future development.

FUNCTIONALITY

Learning Assessment Tools – This tool analyzes the subject's activities, decisions, project performance and self-assessments to determine the subject's competency and learning level achieved. This work involves the development of the logging ability to collect and record the subject's inputs, and a tool to analyze the results. The SEEA infrastructure has been updated to perform the data logging and collection task. The tool is capable of importing the recorded learner data and performing visualization tasks to help experience designers and instructors understand the learners' performance.

Implemented functionalities are (see Figures 7, 8 and 9):

- Collect and format experience data from EA and process data
- Visualize the data gathered from users' experience
- Display users' decision making data
- Compare one experience against another (historic experience or experts' experience)
- Compare students' data within a class against each other in stack chart



UserName:
UAH-Team6-Stud

Experience:
UAVExp14611099

Project Status:
Weight

User Input:

General Actions:
User Name: UAH-Team6-Student1
Experience Start: 2016/04/19_23:52:32
User Name: UAH-Team6-Student1
2016/04/19_23:55:54: Submit
Recommendation
Sim Args:

End Sim Args

Experience Status:
Finished

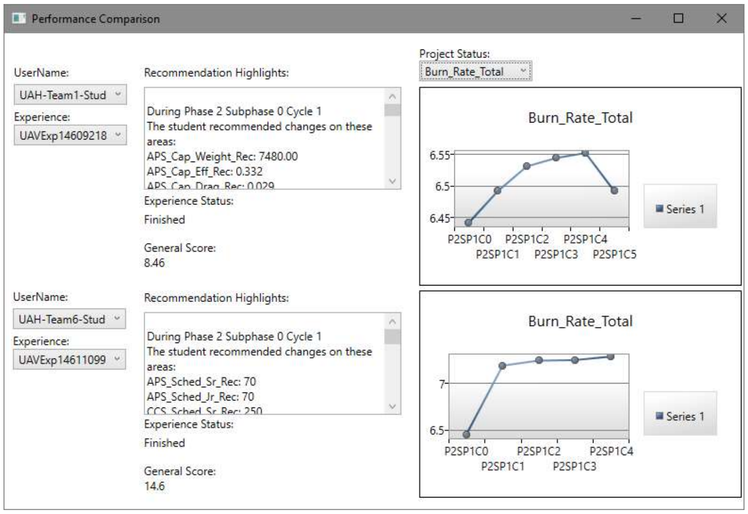
General Score:
14.6

Recommendation Highlights::

During Phase 2 Subphase 0 Cycle 1
The student recommended changes on these areas:
APS_Sched_Sr_Rec: 70
APS_Sched_Jr_Rec: 70
CCS_Sched_Sr_Rec: 250
CCS_Sched_Jr_Rec: 170
CCS_Quality_Sr_Rec: 100
CCS_Quality_Jr_Rec: 50

During Phase 2 Subphase 0 Cycle 2
The student recommended changes on these areas:

Figure 7 Main View of the Learning Assessment Tool



UserName:
UAH-Team1-Stud

Experience:
UAVExp14609218

Recommendation Highlights:

During Phase 2 Subphase 0 Cycle 1
The student recommended changes on these areas:
APS_Cap_Weight_Rec: 7480.00
APS_Cap_Eff_Rec: 0.332
APS_Cap_Drag_Rec: 0.029

Experience Status:
Finished

General Score:
8.46

Project Status:
Burn_Rate_Total

UserName:
UAH-Team6-Stud

Experience:
UAVExp14611099

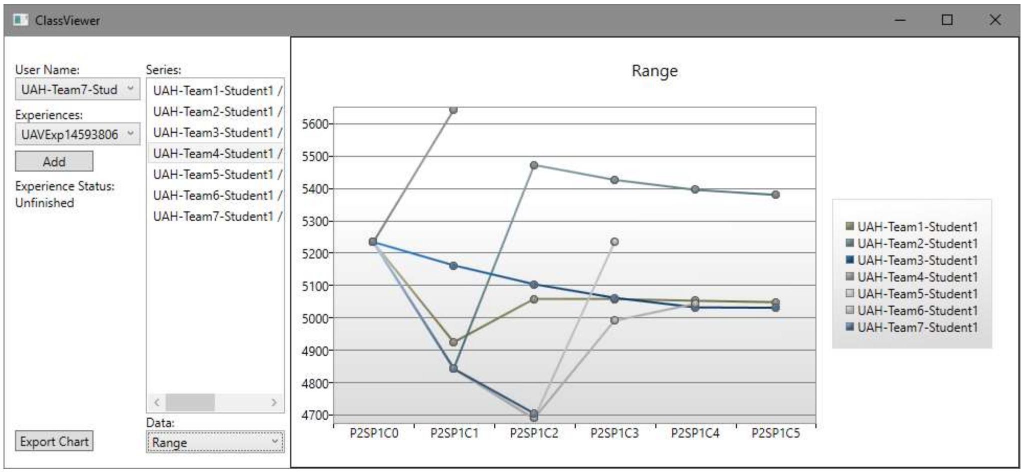
Recommendation Highlights:

During Phase 2 Subphase 0 Cycle 1
The student recommended changes on these areas:
APS_Sched_Sr_Rec: 70
APS_Sched_Jr_Rec: 70
CCS_Sched_Sr_Rec: 250

Experience Status:
Finished

General Score:
14.6

Figure 8 Learning Assessment Tool Performance Comparison



User Name:
UAH-Team7-Stud

Experiences:
UAVExp14593806

Experience Status:
Unfinished

Series:
UAH-Team1-Student1 /
UAH-Team2-Student1 /
UAH-Team3-Student1 /
UAH-Team4-Student1 /
UAH-Team5-Student1 /
UAH-Team6-Student1 /
UAH-Team7-Student1 /

Data:
Range

Figure 9 Learning Assessment Tool Class View

New functionalities for the learning assessment tool focused on aiding the instructors and teachers when he or she is analyzing the students' performance and decision making data.

Unit Testing and Integration Testing

During the development of the Learning Assessment tools, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

Infrastructure Upgrade

During the Part 3 project development, the EA infrastructure was updated by having the EA user interface transitioned from Flash to HTML5. This transition provides the ability to use a wide variety of HTML5 development tools, and greatly improving the ease of rapidly updating the learner interface and its artifacts. HTML5 also facilitates the development of artifacts and user interface updates, enables the SEEA to support a wider range of client devices (iPads, iPhones, etc.) and is critical for Section 508 compliance, which is a requirement for US Government use.

Prior Work

Prior work involved updating the infrastructure, including both the server and client end, to support the use of the tools. The updates include the support to use experience building tools which required updates in the server side code to allow external configurations in the learning experiences. The infrastructure also been provides some of the required updates to allow data recording for the use of the learning assessment tool.

Functionality

The following are the functions that were completed in the Experience Accelerator infrastructure to support the new tools:

- **Update of network protocol from TCP socket to Websocket** - The prior Java TCP socket protocol is not compatible with HTML5 and Javascript. For the HTML5 version to work, a new Websocket protocol based on the HTTP server was developed. The updated server supports the Websocket protocol and thus supports the HTML5 version of the EA.

- **Move certain client functionalities to server side** - The server infrastructure was updated to include multiple client functionalities to make the HTML5 version possible.

- **Conversion of current artifacts** - The current artifacts were converted to support the HTML5 version of EA. This involves the conversion of documents from SWF format to the PDF format. Also some form files were recreated in HTML5 format.

- **Rebuild dialog engine for HTML5** - The dialog engine has been rebuilt and updated for supporting the HTML5 version of the EA. The server dialog parser also was updated to support JSON format of dialog files.

Unit Testing and Integration Testing

During the development of the infrastructure upgrade, unit testing and integration testing were performed. Unit testing was performed when a new feature was added to ensure that it performed as intended. Integration testing was performed, as well, to ensure that changes did not adversely affect existing functionality in the tools. During this process, a variety of defects were identified and remediated.

Evaluation via Experience Development

The evaluation of the tools was conducted with users and potential users of the SEEA. This section presents the results of these evaluations.

Expert Evaluation of Simulation Tools via Wright Brothers Experience

The simulation tools were evaluated by a simulation expert through development of a simulation model to support a new learning experience involving the Wright Brothers and their design and development of new aircraft systems. This is based on an existing assignment used by the Defense Acquisition University (DAU).

Background

The DAU assignment focuses on how the Wright Brothers modified the designs of their existing fliers to meet requirements for a new flier that would be purchased by the U.S. Army. In a five-month development process, they had to meet requirements for range, flight time and speed. The assignment focuses on several facets of what the Wright Brothers needed to accomplish:

- What design decisions should be made to modify the current flier designs to meet new requirements for key performance parameters and technical performance measures?
- How should the work be scheduled and the budget allocated?
- What risks are there, and how should they be managed?

In the assignment, student learners are given a workbook with various artifacts (e.g., original values for KPPs/TPMs and the relationships between them, potential schedules and budgets, etc.). They are then asked to examine these artifacts and answer a series of questions.

Thus, this is a static exercise. The intent here is to use this assignment as the basis for a learning experience in the SEEA. This entails converting the paper assignment into a dynamic experience that plays out over time with changing states and learner input decisions.

SIMULATION MODEL DEVELOPMENT PROCESS

To reduce scope, the focus was limited to the key performance parameters (KPPs) and technical performance measures (TPMs). A simulation model was created to represent the relationship between lift, drag, propulsion and weight. A static relationship diagram for these factors is included in the DAU assignment. This provides clues to the learner that the two factors are critical to improving the range to meet the ranger requirement of 125 miles are (i) decreasing weight and (ii) increasing lift or decreasing drag. The Wright Brothers decreased the weight of the flyer that they used as the starting point, and they also increased its wing area (providing additional lift).

A simulation model was created using the Simulation Modeler that incorporates lift, drag, propulsion efficiency and weight. The simulation model uses the Breguet range equation for propeller aircraft as the computational engine for the model. The model is depicted in Figure 10.

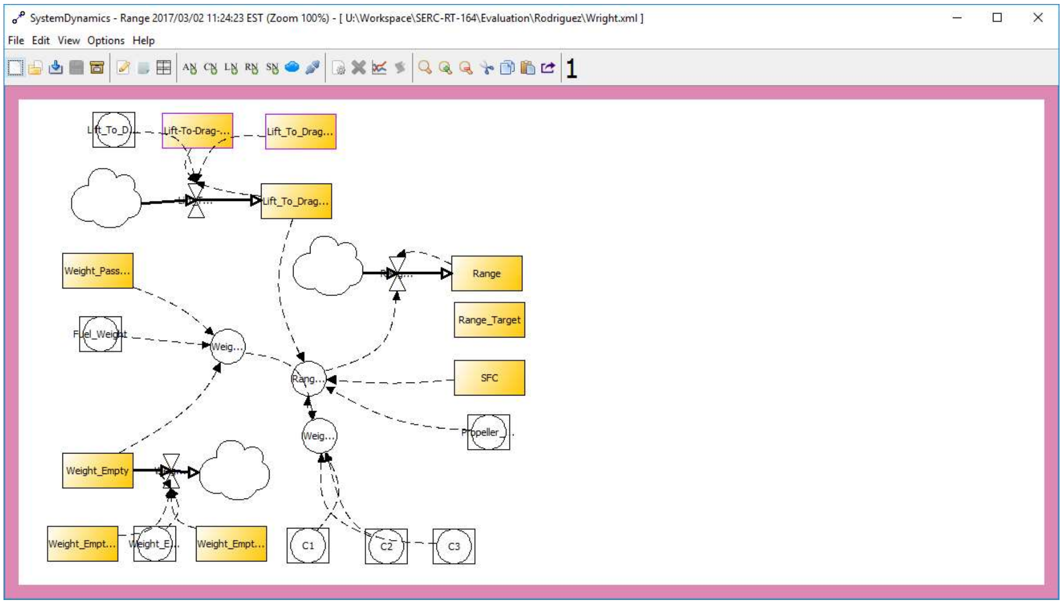


Figure 10. Wright Brothers KPP/TPM model

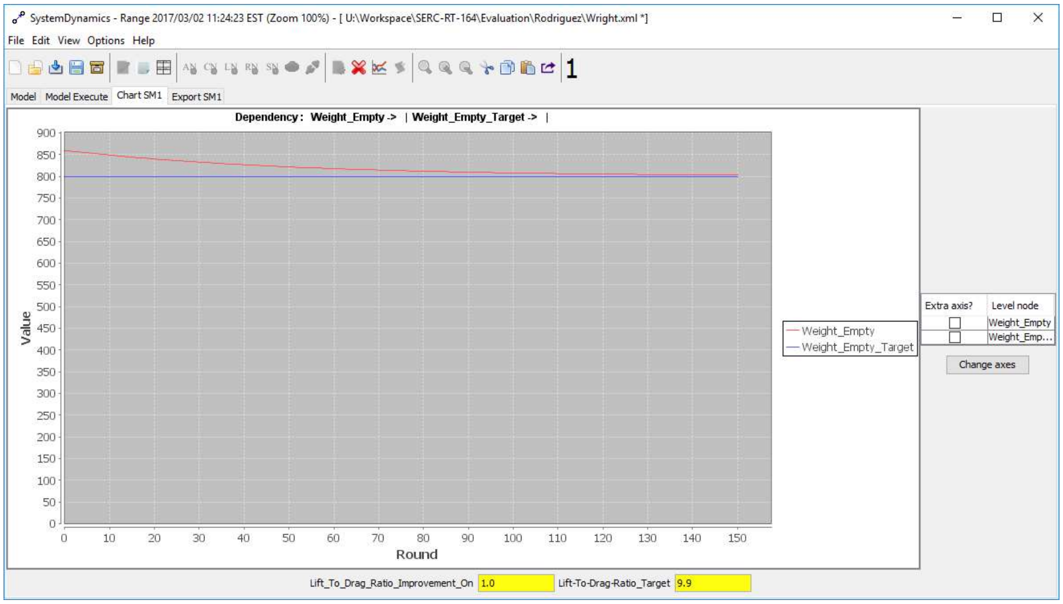
The variable Range tracks the current estimate for range of the flyer. The variable Range_Target is set at 125 miles (i.e., the requirement). The specific fuel consumption (SFC) and propeller efficiency (Propeller_Efficiency) affect range, but it is assumed that these items are not subject to being modified.

The equation considers the empty weight of the aircraft (Weight_Empty), the weight of fuel (Weight_Fuel), and the weight of passengers (Weight_Passengers). The empty weight starts at

860 lbs. There is a requirement to carry 350 lbs. in passenger weight. Fuel weight is assumed to be 500 lbs. The learner can seek to reduce the empty weight of the flyer. Similarly, the lift-to-drag ratio (Lift_To_Drag_Ratio) can be modified by the learner to represent the effect of increased wing surface area. It starts at 8.30. (Note: the ratio is dimensionless.)

The simulation model has a set of two variables for each of these decisions. For decreased empty weight, the learner activates the Empty_Weight_Improvement_On variable, and also sets the Weight_Empty_Target variable to the desired weight. The Empty_Weight_Improvement_On variable is a binary variable set originally to zero (i.e., no weight improvement effort underway). Setting it to one enables the weight improvement. For increased lift-to-drag ratio, the learner sets the Lift_To_Drag_Ratio_Improvement_On variable, and also sets the Lift_To_Drag_Ratio_Target variable to the desired value. The Lift_To_Drag_Ratio_Improvement_On variable is a binary that represents whether there is an active effort to improve the lift-to-drag ratio.

Using the Sim Tuner, we can see the effect of enabling a weight improvement program for the flyer with a target empty weight of 800 lbs. It is assumed that weight reductions are found gradually over the course of the five-month development program (150 days on the x-axis). This is shown in Figure 11.



SystemDynamics - Range 2017/03/02 11:24:23 EST (Zoom 100%) - [U:\Workspace\SERC-RT-164\Evaluation\Rodriguez\Wright.xml *]

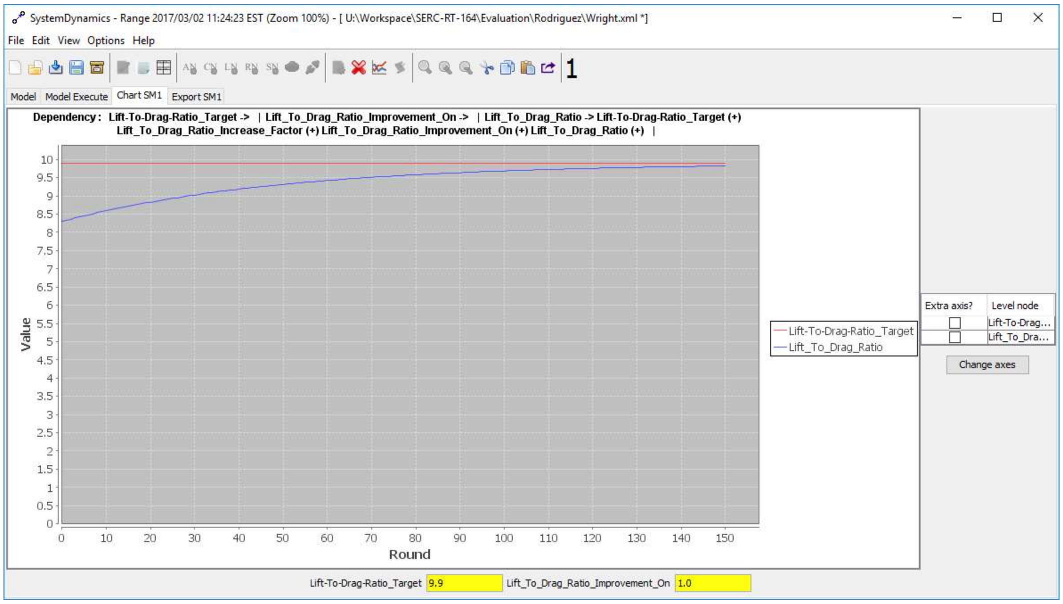
Dependency: Weight_Empty -> | Weight_Empty_Target -> |

Extra axis?	Level node
□	Weight_Empty
□	Weight_Emp...

Lift_To_Drag_Ratio_Improvement_On 1.0 Lift-To-Drag-Ratio_Target 9.9

Figure 11. Weight reduction program

Similarly, we can see the effect of enabling a lift-to-drag ratio improvement program over the five-month program in Figure 12. The ratio starts at 8.30, and the learner sets a target of 9.90.

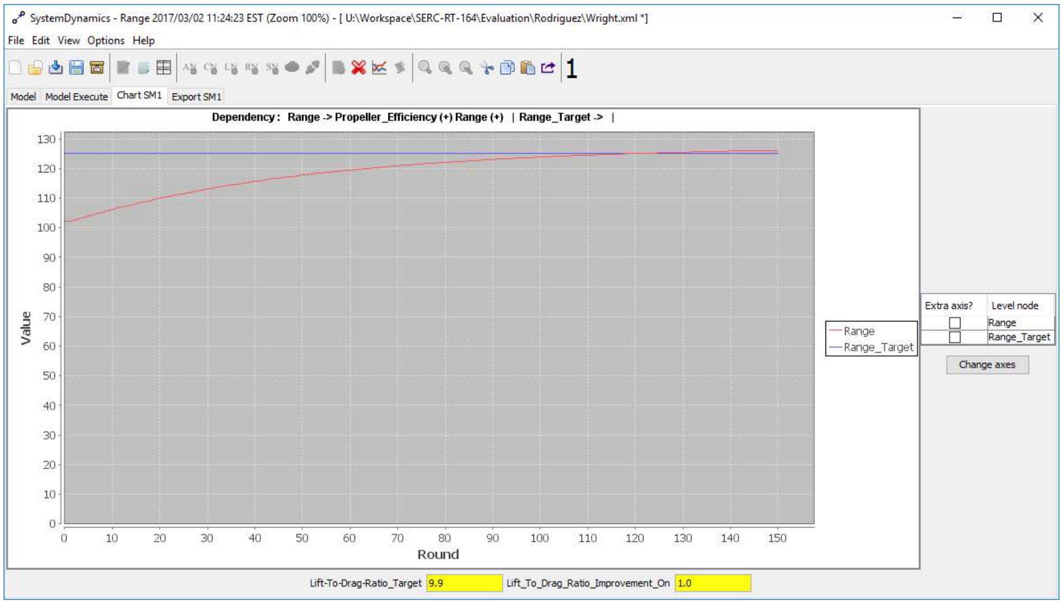

SystemDynamics - Range 2017/03/02 11:24:23 EST (Zoom 100%) - [U:\Workspace\SERC-RT-164\Evaluation\Rodriguez\Wright.xml *]

Dependency: Lift-To-Drag-Ratio_Target -> | Lift_To_Drag_Ratio_Improvement_On -> | Lift_To_Drag_Ratio -> Lift-To-Drag-Ratio_Target (+) Lift_To_Drag_Ratio_Increase_Factor (+) Lift_To_Drag_Ratio_Improvement_On (+) Lift_To_Drag_Ratio (+) |

Lift-To-Drag-Ratio_Target 9.9 Lift_To_Drag_Ratio_Improvement_On 1.0

Figure 12. Lift-to-drag ratio improvement program

Of course, the learner is concerned with meeting the overall range requirement of 125 miles. If we combine the weight improvement program and the lift-to-drag ratio improvement program, we can see the improvement in range over the five-month program in Figure 13. The learner has just exceeded the requirement.

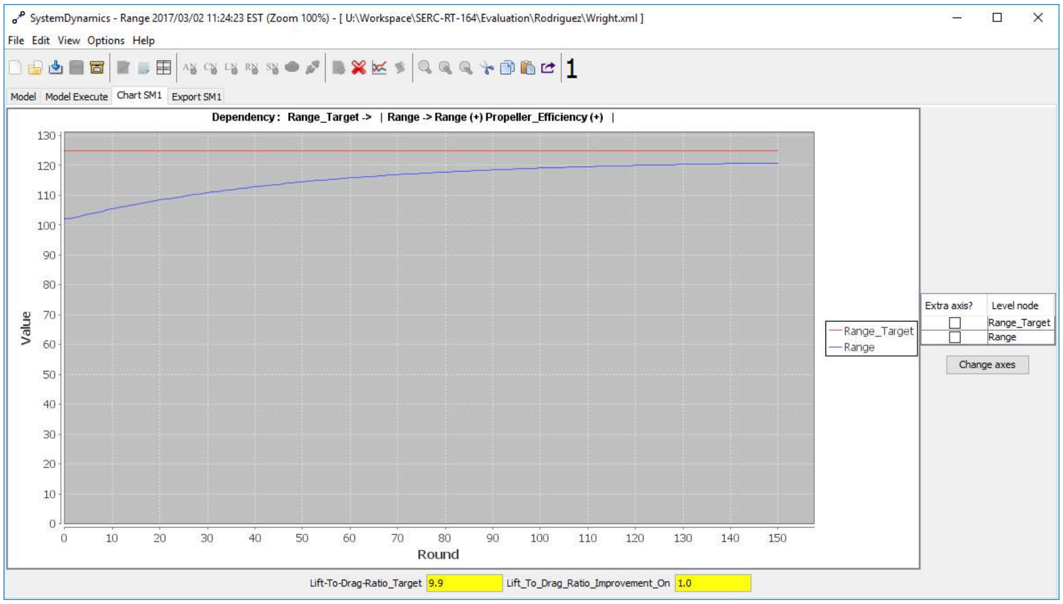

SystemDynamics - Range 2017/03/02 11:24:23 EST (Zoom 100%) - [U:\Workspace\SERC-RT-164\Evaluation\Rodriguez\Wright.xml *]

Dependency: Range -> Propeller_Efficiency (+) Range (+) | Range_Target -> |

Lift-To-Drag-Ratio_Target 9.9 Lift_To_Drag_Ratio_Improvement_On 1.0

Figure 13. Range improvement

We can ask what happens if the learner enables only one of these programs, say the lift-to-drag ratio improvement program. Figure 14 shows the resulting range improvement without the weight improvement. Obviously, the improvement effort falls short.


SystemDynamics - Range 2017/03/02 11:24:23 EST (Zoom 100%) - [U:\Workspace\SERC-RT-164\Evaluation\Rodriguez\Wright.xml]
File Edit View Options Help
1
Model
Model Execute
Chart SM1
Export SM1
Dependency : Range_Target -> | Range -> Range (+) Propeller_Efficiency (+) |
Value
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
Round
Range_Target
Range
Change axes

Extra axis?	Level node
□	Range_Target
□	Range

Lift-To-Drag-Ratio_Target 9.9 Lift_To_Drag_Ratio_Improvement_On 1.0

Figure 14. Range improvement without weight improvement

In this example model, the learner was able to modify both TPMs so that the overall KPP requirement was met. It is possible that the flyer is such that the lift-to-drag ratio cannot exceed a certain amount. The model can be designed so that this constraint is in place.

Note that the designer uses the Sim Builder to build the model and the Sim Tuner to test its behavior. In the SEEA, the learner would enter inputs via a recommendation form, and the SEEA application would change the values of the variables input into the simulation model.

Overall, the tools proved useful in developing a model to support a Wright Brothers experience in the span of approximately two hours.

NOVICE EVALUATION OF SIMULATION TOOLS VIA WRIGHT BROTHERS EXPERIENCE

The tools were also evaluated by a Defense Acquisition University (DAU) instructor who was not familiar with simulation or system dynamics. The tools and the model file for the Wright Brothers experience were provided, as well as documentation. This section provides details of the results of the review. Summarizing, the tools were relatively straightforward in understanding how they work, given the limited time spent in training. However, there is a gap between this and the user being able to develop a new model or extend an existing one without training and/or additional mentoring. The user makes several recommendations in this regard for deployment at DAU.

1. Evaluation goals and objectives

The goal of this evaluation is to evaluate the use of the SystemDynamics tool.
The objective of this evaluation is to evaluate the instructions and ease of use to develop experience supporting charts demonstrating decision impacts.

2. **Initial product assessments**

   The product did appear easy to use with the one-on-one guidance from an expert user. Once the novice user was on their own the development was difficult. A significant amount of time is required to obtain enough experience to properly use the SystemDynamics tool. The tutorial material available explains the use of the SystemsDynamic tool, but there is not a reference document provided to explain the product interface between the SystemsDynamic output and the Systems Engineering Experience Accelerator. The tool will be very difficult to use by individuals without modeling or programming experience.

3. **Additional products or currently deployed solutions within the sponsor's environment worth considering**

   N/A

4. **Considerations/requirements for the sponsor's environment**

   The experience developer responsibilities require additional consideration. The current DAU ENG301 course has a significant amount of faculty responsibility in executing the course offering and it is unreasonable to expect faculty members to invest the time and labor necessary to become an expert user.

   We may consider developing a specific SEEA team at DAU to manage and develop experiences. It is unreasonable to expect a single point of contact to maintain these responsibilities.

5. **Evaluation criteria and the test plan**

   Step 1: Expert user provided novice user with training/tutorial materials. The following training materials were readily available, but it is recommended that specific tailored training materials be developed to aid the DAU user to understand the use and interfaces of the SystemsDynamic tool.

   60 Minutes of reading and following examples
   System Dynamic Simulation Builder documentation: Easy to follow instructions

   48 Minutes
   Youtube tutorial
   https://www.youtube.com/watch?v=_ArCpFZVVas

   10 Minutes
   Anylogic tutorial (commercial software)
   https://en.wikibooks.org/wiki/Simulation_with_AnyLogic/System_Dynamics

   11 + 10 +15 Minutes = 36 Minutes
   iThink tutorial (commercial software)
   https://www.youtube.com/watch?v=V3pPQk5Opc8
   https://www.youtube.com/watch?v=jehYN3yULHo
   https://www.youtube.com/watch?v=_jEopppYOdE

Semester Course
MIT course
https://ocw.mit.edu/courses/sloan-school-of-management/15-871-introduction-to-system-dynamics-fall-2013/

Step 2: Expert user provided one-on-one time using Webex share screen with the novice user. This guidance was easy to follow. Basic examples were explored and found to be informative.

Step 3: The expert user provided the novice user a SystemsDynamic product, Range, to evaluate and recommended to develop a similar product. After approximately four hours of reviewing the Range product and attempting to develop a Weight product to interface with the Range product, the novice user was unsuccessful.

Simulation Tools Evaluation via Systems Thinking Application

Finally, the simulation tools were evaluated by a Ph.D student at Stevens Institute of Technology, who is using the SEEA for his research and is building an experience to evaluate systems thinking skills. Overall, he expressed interest in using the Simulation Modeler tools for his research. The student provided detailed suggestions and recommendations mainly related to usability and navigation. The details are provided in Appendix C, along with updates for those that have been addressed. The recommendations fall into three broad categories summarized below.

- User interface – most recommendations focused on improvements to the user interface, such as zoom, context menus for model elements, location of menu-bar functions, resizing the screen, labels for various items, resizing model elements, etc.

- Model editing functions – several recommendations addressed the editing of models, in particular deleting of dependent model elements and selection of multiple model elements.

- New/upgraded features/functions – two features and functions were recommended: separating the validate function out from model read/save, and enriching the sub-model function so that sub-models can be organized into graphical "maps" showing their relationships with the ability to zoom/unzoom and shrink/enlarge sub-models and model elements. The latter is particularly challenging, but could be a very useful addition to the tools.

Due to time limitations, most of these could not be addressed in this research period, but serve as recommendations for future work.

EVALUATION OF EXPERIENCE BUILDING TOOLS

The evaluation of the experience building tools was conducted with users and potential users of the SEEA. This section presents the results of evaluations of tool use in the development of three new experiences: UK MoD HMS Tempest, Project Robot, and Wright Brothers.

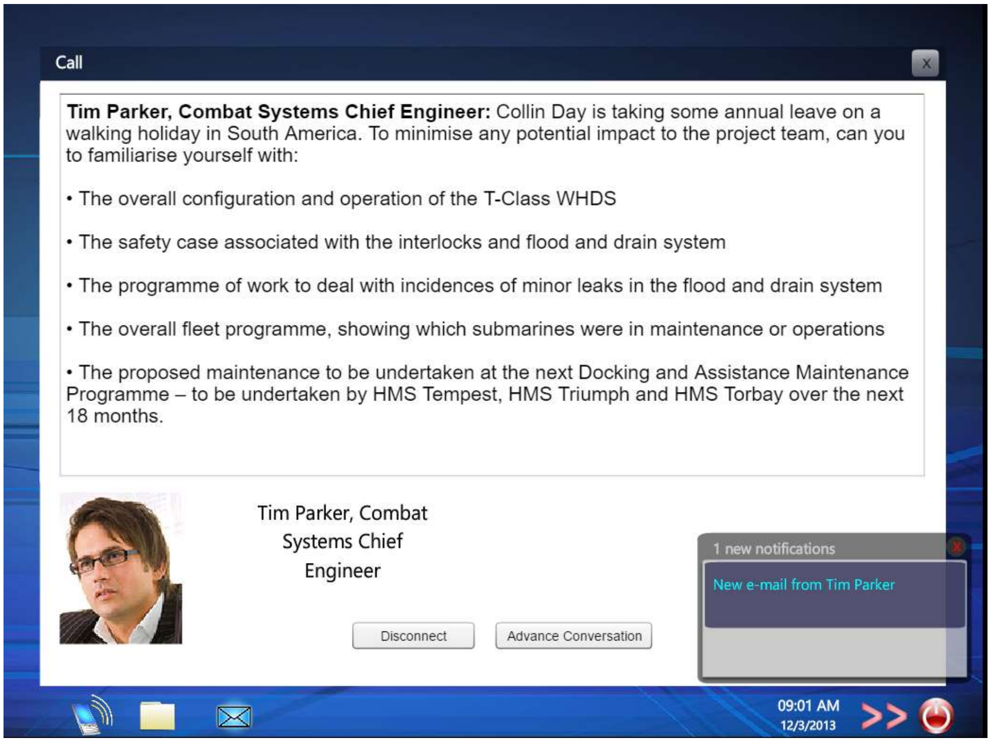
EXPERIENCE BUILDING TOOLS EVALUATION VIA UK MOD HMS TEMPEST EXPERIENCE

The UK MoD HMS Tempest Experience focuses on a submarine maintenance scenario where the learner is responsible for finding issues with the submarine by communicating with other personnel in the team. Failing to communicate with peers in a timely fashion results in a catastrophic outcome. Since this experience design does not require a simulation engine, experience creation required only the utilization of the experience building tools. The creation of this experience demonstrated how the tools can be used to help the creation of new experiences.

The experience is divided into five phases:

- New User Orientation Phase – Welcomes the learner and provides the learner background information about the situation.
- Start of Experience Phase – The experience starts with a phone call indicating a potential threat.
- Investigation Phase – The learner investigates the potential issues by communicating with peers in a timely and appropriate manner.
- Reporting phase – The learner reports to the chief engineer about the investigation and makes recommendations.
- Debriefing Phase – Depending on the learner’s actions, different outcomes will result and the learner receives feedback about his/her experience performance.

For the new user orientation phase, an email is sent to the learner requiring him/her to read through five technical documents and proceed. Artifact Integrator was used to convert and integrate the PDF formatted documents. The email was composed in Artifact Integrator and then referenced from Event Editor where the email event was created. One limitation of the tools is that they currently do not support the creation of user interface elements, therefore some coding efforts were needed to create a proceed button on the user interface (UI). Figure 15 shows a phone call event in new user orientation phase.



Tim Parker, Combat Systems Chief Engineer: Collin Day is taking some annual leave on a walking holiday in South America. To minimise any potential impact to the project team, can you to familiarise yourself with:

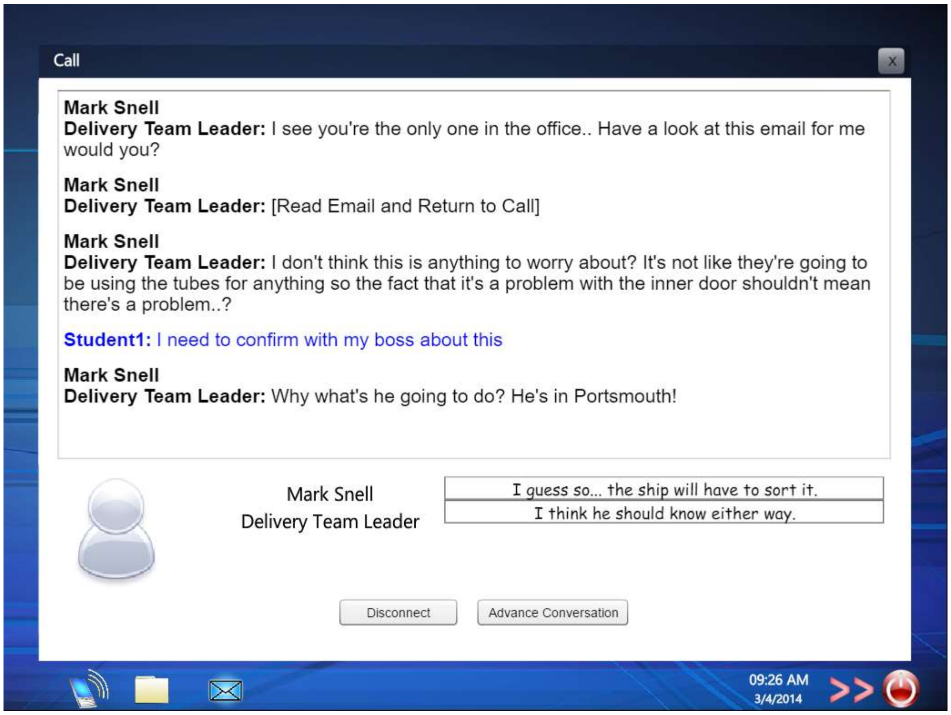
- The overall configuration and operation of the T-Class WHDS
- The safety case associated with the interlocks and flood and drain system
- The programme of work to deal with incidences of minor leaks in the flood and drain system
- The overall fleet programme, showing which submarines were in maintenance or operations
- The proposed maintenance to be undertaken at the next Docking and Assistance Maintenance Programme – to be undertaken by HMS Tempest, HMS Triumph and HMS Torbay over the next 18 months.

Tim Parker, Combat Systems Chief Engineer

1 new notifications

New e-mail from Tim Parker

Figure 15 HMS Tempest Experience Setup Phase



Mark Snell
Delivery Team Leader: I see you're the only one in the office.. Have a look at this email for me would you?

Mark Snell
Delivery Team Leader: [Read Email and Return to Call]

Mark Snell
Delivery Team Leader: I don't think this is anything to worry about? It's not like they're going to be using the tubes for anything so the fact that it's a problem with the inner door shouldn't mean there's a problem..?

Student1: I need to confirm with my boss about this

Mark Snell
Delivery Team Leader: Why what's he going to do? He's in Portsmouth!

Mark Snell
Delivery Team Leader

I guess so... the ship will have to sort it.

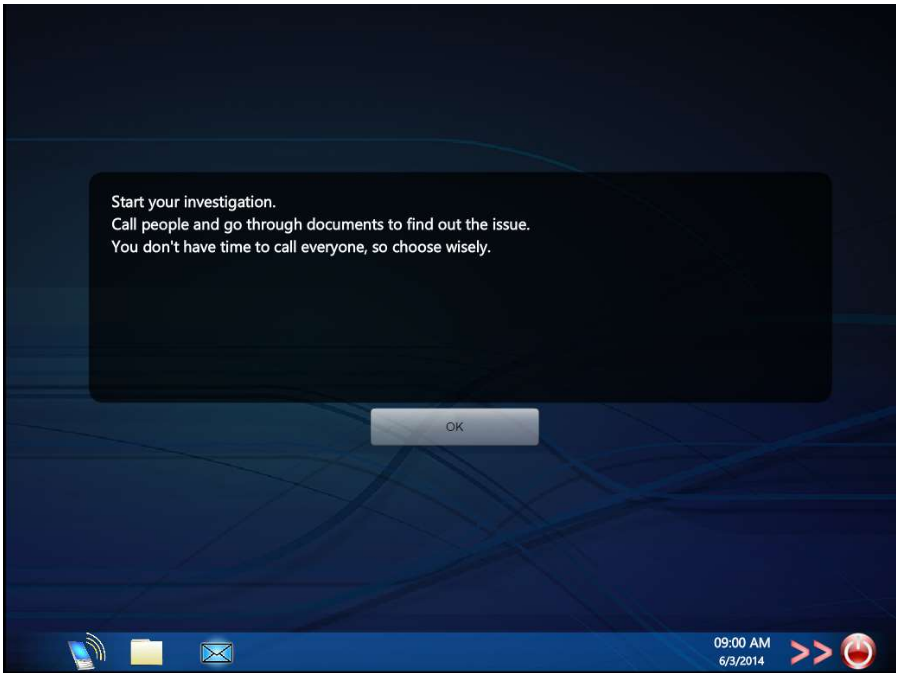
I think he should know either way.

Figure 16 HMS Tempest Experience Start Phase

For the start of experience phase, the commercial tool *Chat Mapper* was used to create the phone call from the delivery team leader asking for recommendations on whether a situation should be ignored. Event Editor was used to create the phone call and email events for this phase. Figure 16 shows the dialogue between the learner and delivery team leader during the start of experience phase.

Figure 17 shows the start of the investigation phase where a message directing the tasks for the learner is displayed. For the investigation phase, Chat Mapper was used to create phone call dialogs between the learner and NPC. Event Editor was used to create email and NPC events. These events are variable based and triggered by user actions.

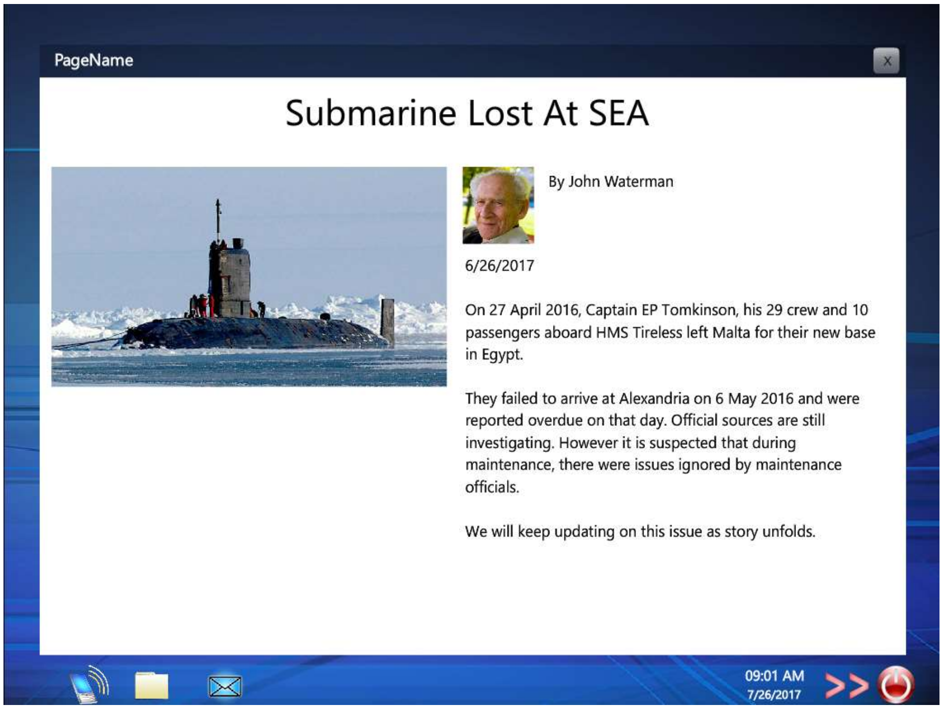
The creation of the reporting phase involves composing dialogues. The available dialogue options are based on the results from the investigation phase. Event Editor and Chat Mapper are utilized in creating this phase.



Start your investigation.
Call people and go through documents to find out the issue.
You don't have time to call everyone, so choose wisely.

Figure 17 HMS Tempest Experience Investigation Phase

The debriefing phase requires the use of Artifact Integrator and Event Editor to create feedback messages based on user performance. Figure 18 shows the final news briefing page for the experience where the learner will learn the outcomes of their decision making process.



Submarine Lost At SEA

By John Waterman

6/26/2017

On 27 April 2016, Captain EP Tomkinson, his 29 crew and 10 passengers aboard HMS Tireless left Malta for their new base in Egypt.

They failed to arrive at Alexandria on 6 May 2016 and were reported overdue on that day. Official sources are still investigating. However it is suspected that during maintenance, there were issues ignored by maintenance officials.

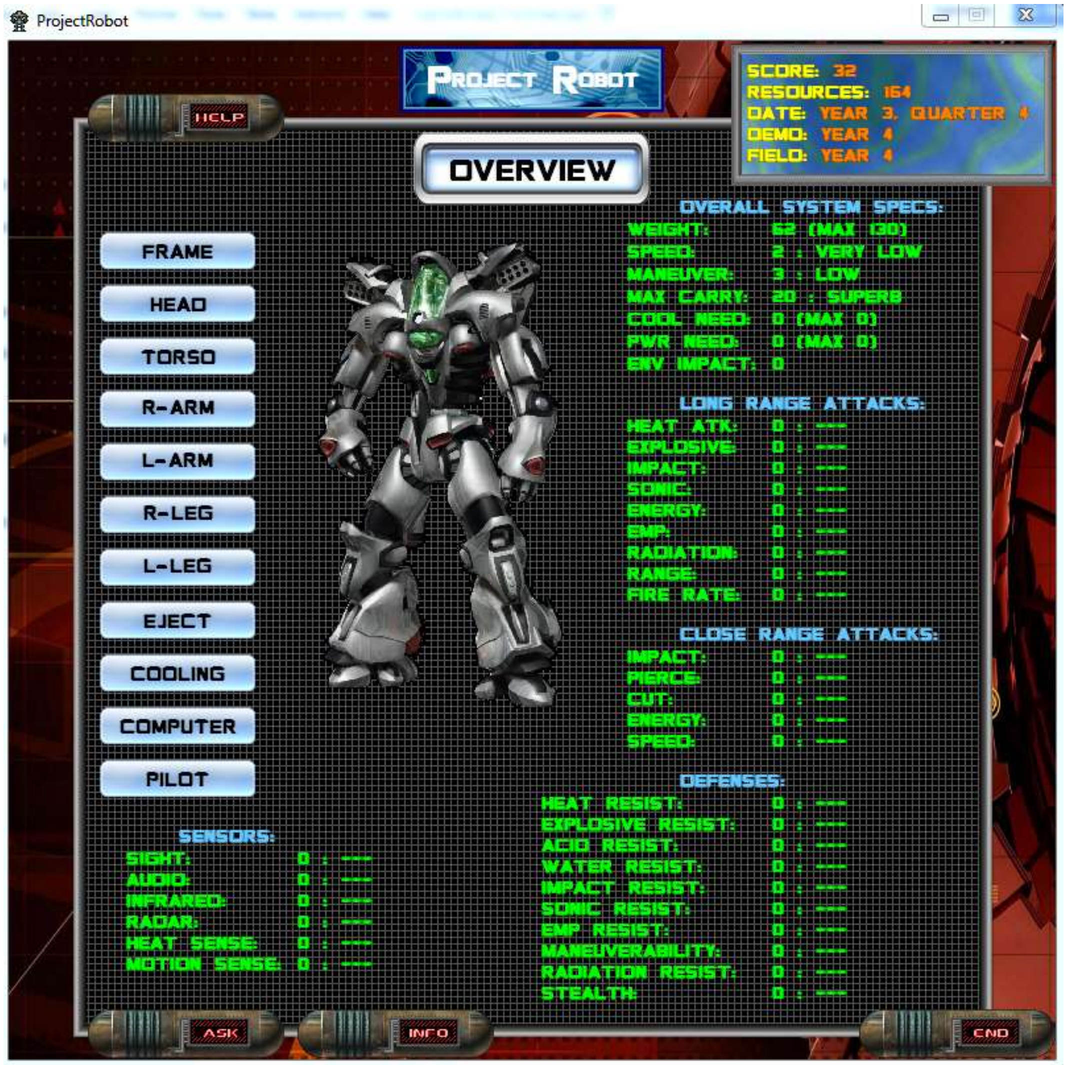
We will keep updating on this issue as story unfolds.

Figure 18 HMS Tempest Experience Debriefing Phase

After the creation of events and artifacts, Phase Editor is used to connect the phases together and assign events and available NPCs.

Experience Building Tools Evaluation via Project Robot Experience

The experience building tools were also evaluated by two masters students at Stevens Institute of Technology, who are using the SEEA for their special problem course. Students are building an Experience Accelerator Challenge Experience, focusing on decision making and resource allocation, which is based on *Project Robot*, developed by Ross Arnold which was a co-winner of the Stevens School of Systems and Enterprises Experience Accelerator contest in 2010. The overall goal of the learner experience is to create a robot with medium speed, high maneuverability, with high defense, and medium to low weight. The experience is populated with NPC dialogues, random events, available documentations and stakeholder updates. These experience elements were developed using the experience building tools. Figure 19 shows the design concept of the experience user interface.



SCORE: 32
RESOURCES: 164
DATE: YEAR 3, QUARTER 4
DEMO: YEAR 4
FIELD: YEAR 4

OVERALL SYSTEM SPECS:
WEIGHT: 62 (MAX 130)
SPEED: 2 : VERY LOW
MANEUVER: 3 : LOW
MAX CARRY: 20 : SUPERB
COOL NEED: 0 (MAX 0)
PWR NEED: 0 (MAX 0)
ENV IMPACT: 0

LONG RANGE ATTACKS:
HEAT ATK: 0 : ---
EXPLOSIVE: 0 : ---
IMPACT: 0 : ---
SONIC: 0 : ---
ENERGY: 0 : ---
EMP: 0 : ---
RADIATION: 0 : ---
RANGE: 0 : ---
FIRE RATE: 0 : ---

CLOSE RANGE ATTACKS:
IMPACT: 0 : ---
PIERCE: 0 : ---
CUT: 0 : ---
ENERGY: 0 : ---
SPEED: 0 : ---

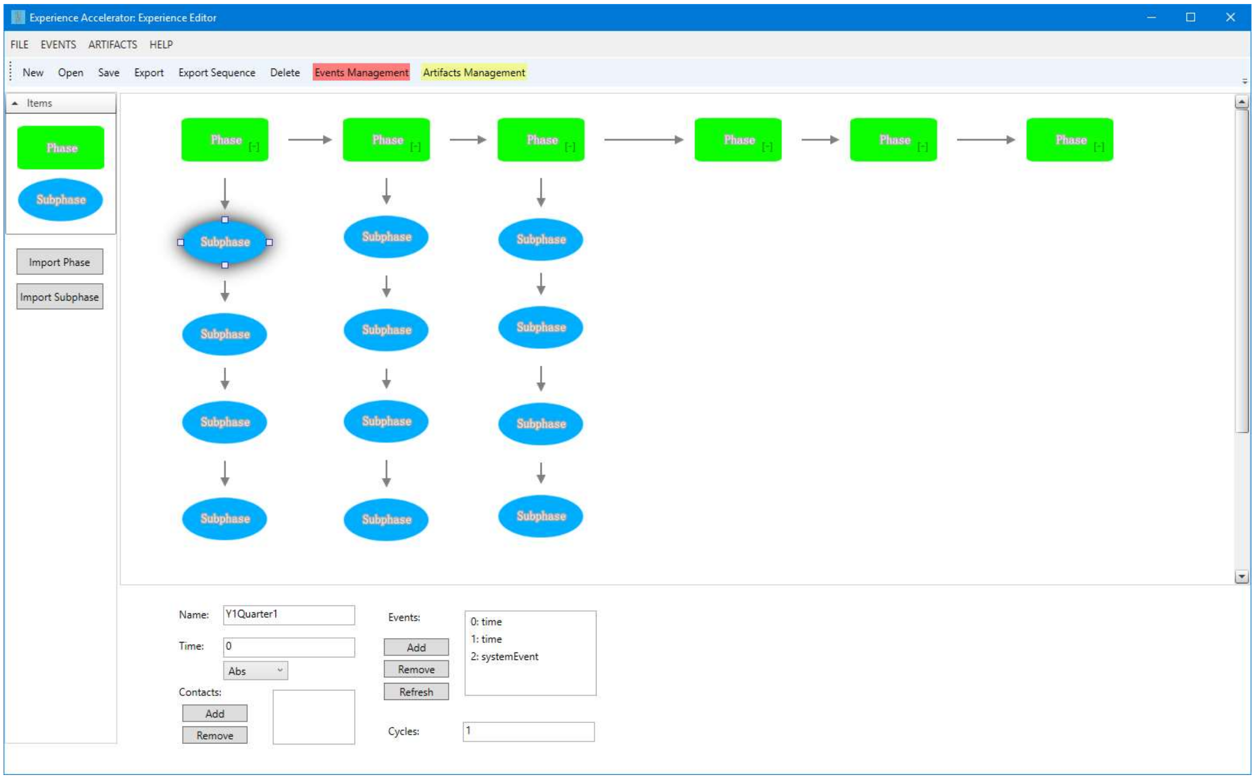
DEFENSES:
HEAT RESIST: 0 : ---
EXPLOSIVE RESIST: 0 : ---
ACID RESIST: 0 : ---
WATER RESIST: 0 : ---
IMPACT RESIST: 0 : ---
SONIC RESIST: 0 : ---
EMP RESIST: 0 : ---
MANEUVERABILITY: 0 : ---
RADIATION RESIST: 0 : ---
STEALTH: 0 : ---

SENSORS:
SIGHT: 0 : ---
AUDIO: 0 : ---
INFRARED: 0 : ---
RADAR: 0 : ---
HEAT SENSE: 0 : ---
MOTION SENSE: 0 : ---

Figure 19 Main User Interface for Project Robot (Source: Project Robot)

The Project Robot was designed to be the simulation of a 10-year project. The experience is divided into seven phases, with the first three phases having four subphases each. Shown in Figure 20, the experience structure was implemented using the Phase Editor tool. Phases 1-3 are mapped to years 1-3 with each subphase representing a quarter. Phases 4-6 are mapped to years 4-9 with each phase representing a two year span. The final phase is year 10.

Available documentations and stakeholder updates are designed using the artifact management tool. PDF files are imported using the tool and assigned to different phases. Stakeholders will act as NPCs in the experience through phone calls. NPCs are designed using the NPC creation function in the artifact management tool, and the dialogues are designed using the third-party tool Chat Mapper.


Experience Accelerator: Experience Editor
FILE EVENTS ARTIFACTS HELP
New Open Save Export Export Sequence Delete Events Management Artifacts Management
Items
Phase
Subphase
Import Phase
Import Subphase
Phase [-]
Subphase
Add
Remove
Refresh

Name: Y1Quarter1

Time: 0

Abs

Contacts:

Events:

0: time
1: time
2: systemEvent

Cycles: 1

Figure 20 Project Robot Experience Structure Design

The experience is still under development at this point. Overall the tools received positive feedbacks from the students. The students also provided recommendations for future improvement. Due to time constraints, most of the recommended changes will be addressed in future development. The following are the detailed feedbacks from the students:

> *"After using this tool just for a few hours, I have some feedback on possible improvements. This tool is very straightforward and easy to use but there are a few things that could be updated from a user standpoint. While editing in the phase section, if a phase/subphase is previously selected, the user must click off to a blank space before selecting another phase/subphase to edit. Also the phases will get highlighted and stay highlighted simply just by dragging your mouse past the phase/subphase.*
>
> *Another update would be to make the phase name appear to the user visually on the phase block rather than just the word "phase" or "subphase" for each one.*
>
> *Lastly a help section should be added which details how each section should be used properly and what each variable means. This includes detailing how to use the events management as well as the artifacts section. The different types of events are very confusing without documentation explaining them. Overall this toolbox makes it much easier for a non-programmer to build a significant portion of the game."*

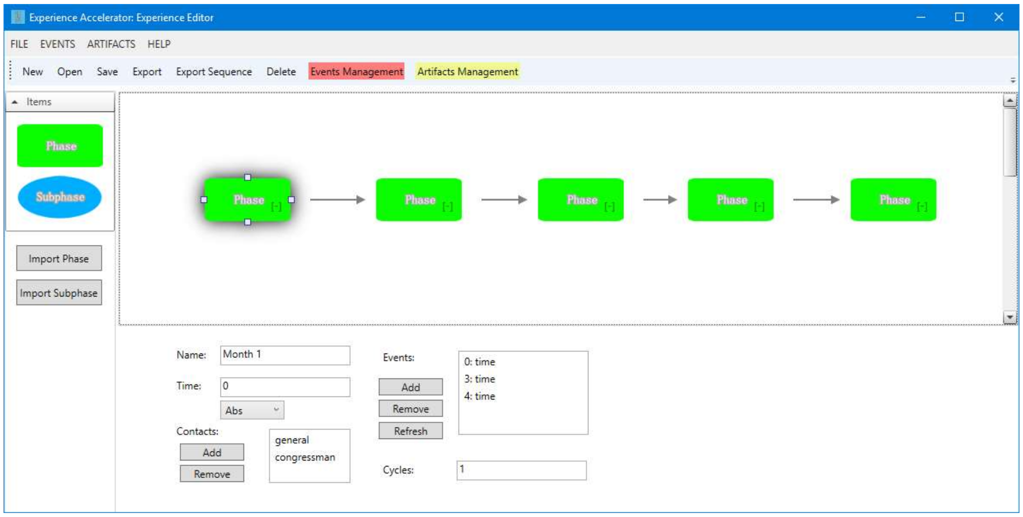
EXPERT EVALUATION OF EXPERIENCE BUILDING TOOLS VIA WRIGHT BROTHERS EXPERIENCE

The experience building tools were also evaluated through development of a new learning experience involving the Wright Brothers and their design and development of new aircraft systems. This is based on an existing assignment used by the Defense Acquisition University (DAU). The DAU assignment focuses on how the Wright Brothers modified the designs of their existing fliers to meet requirements for a new flier that would be purchased by the U.S. Army. The experience lasts five-month in-game time for the development process.

In the assignment, student learners are given a workbook with various artifacts (e.g., original values for KPPs/TPMs and the relationships between them, potential schedules and budgets, etc.). They are then asked to examine these artifacts and answer a series of questions.

As a five-month long experience, the whole story can be divided into five phases with each phase representing one month of development time for the Wright brothers. Student learners will be required to go through the artifacts and then make decisions on design changes. At the beginning of each phase, learners can retrieve important information about the project and the army's requirements by examining the updated artifacts, NPC dialogues and flying machine status. They then need to make decisions on possible changes to be made to the project. After each phase, the learner's input will be input into the simulator which will then be updated to the new status of the project and attributes of the flying machine.

The Phase Editor tool can be used to create the baseline structure of the experience. Figure 21 shows the design of the experience implemented using Phase Editor tool. The experience consists of five development phases, with each phase lasting one month. The available NPCs and events are set as well.



Name: Month 1

Time: 0

Abs

Contacts: general, congressman

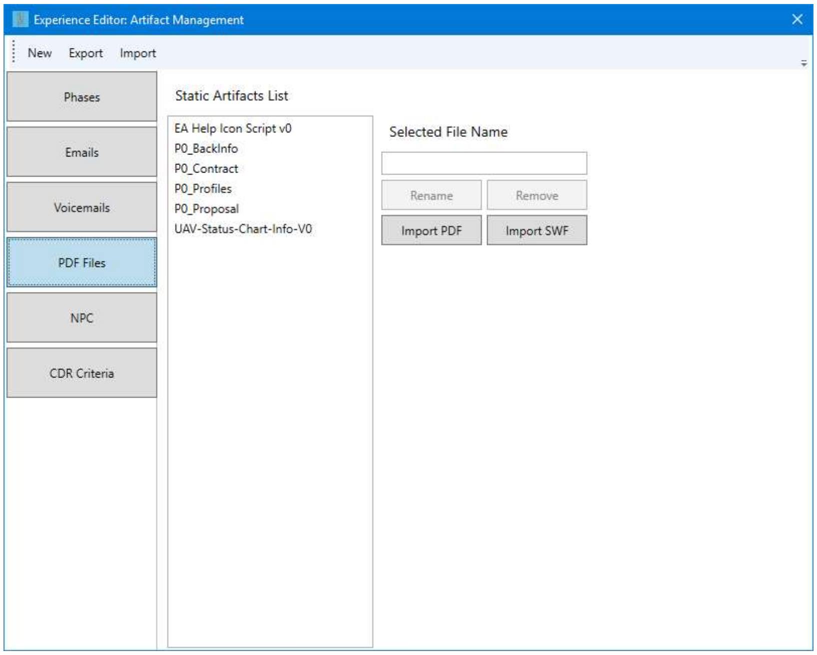
Events: 0: time, 3: time, 4: time

Cycles: 1

Figure 21 Baseline Structure of the Wright Brothers Experience

For creating and importing artifacts, the Artifact Management tool can be used to populate the experience with emails, voicemails, PDF files and non-player characters. In the case of this experience which predates computers and the internet, voicemail features are not used and the email feature can be treated as postal mails or telegrams.

The Artifact Management tool can be used to import the artifacts into the EA format, which can be made available to the students at certain stage of the experience. Figure 22 shows the tool for importing the PDF documents for Wright Brothers Experience.

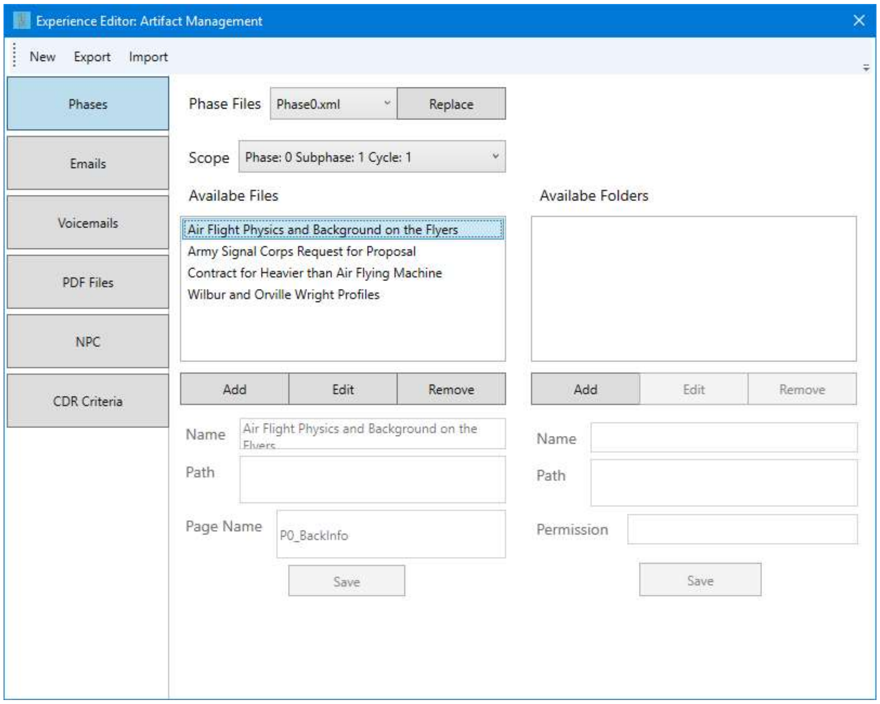


Static Artifacts List

EA Help Icon Script v0
P0_BackInfo
P0_Contract
P0_Profiles
P0_Proposal
UAV-Status-Chart-Info-V0

Figure 22 Import PDF Artifacts using Artifact Management Tool

The documents are then assigned to Phase 0 of the experience using the Phases section in the Artifact Management tool. Figure 23 shows the documents assigned to Phase 0 using the phase files function.



Phase Files: Phase0.xml

Scope: Phase: 0 Subphase: 1 Cycle: 1

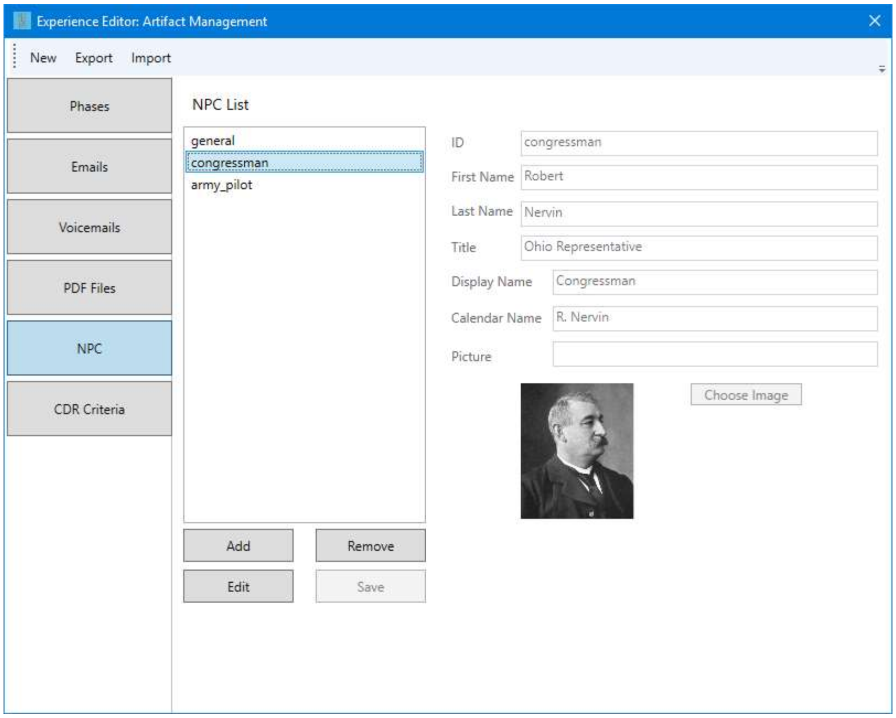
Air Flight Physics and Background on the Flyers
Army Signal Corps Request for Proposal
Contract for Heavier than Air Flying Machine
Wilbur and Orville Wright Profiles

Name: Air Flight Physics and Background on the Flyers

Page Name: P0_BackInfo

Figure 23 Assign Artifacts to Phases using Artifact Management Tool

To implement the dialogues with non-player characters (NPCs), the NPCs should be designed first. Figure 24 demonstrates the NPC creation for the experience. The created NPCs are General Frank Lahm, Congressman Robert Nervin, and Lieutenant Thomas Selfridge. The tool supports the configuration of NPCs’ IDs, Names, Titles and photos.



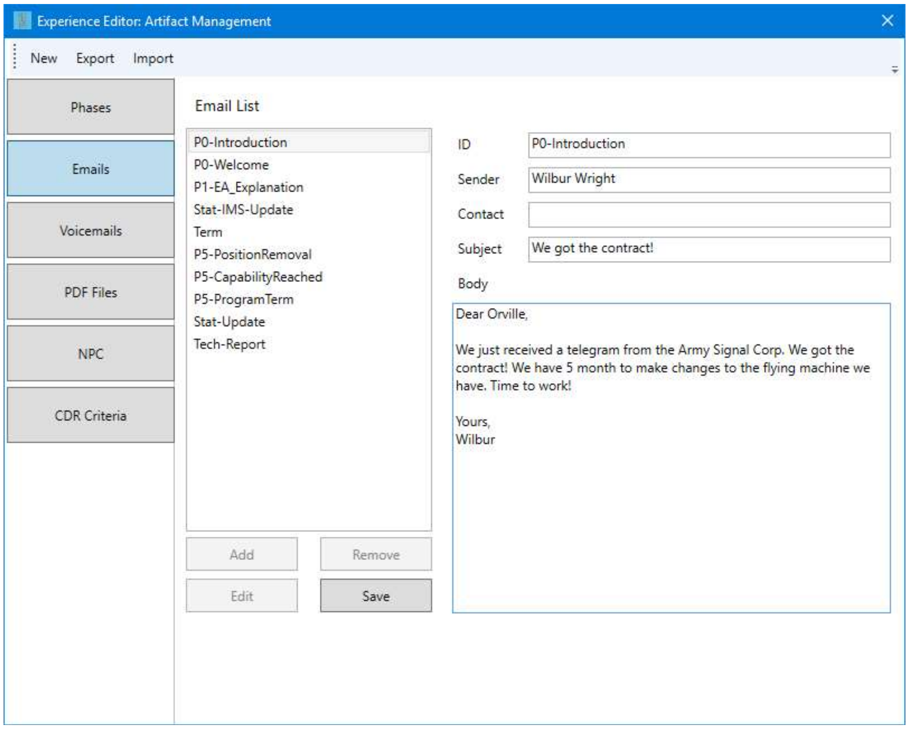
general
congressman
army_pilot

ID: congressman
First Name: Robert
Last Name: Nervin
Title: Ohio Representative
Display Name: Congressman
Calendar Name: R. Nervin
Picture:

Figure 24 NPC Creation using Artifact Management Tool

Telegrams or postal mails can be created in the Email section of the Artifact Management tool. Figure 25 is the screenshot of creating a postal mail from Wilbur Wright.

After the implementation of artifacts, the next step in experience design is event designing where the Event Management tool was used to create/mange events. Figure 26 presents the event creation of a mail event using the Event Management tool. These events can then be added into phases and subphases using the Phase Editor tool.



Email List

- P0-Introduction
- P0-Welcome
- P1-EA_Explanation
- Stat-IMS-Update
- Term
- P5-PositionRemoval
- P5-CapabilityReached
- P5-ProgramTerm
- Stat-Update
- Tech-Report

ID: P0-Introduction

Sender: Wilbur Wright

Contact:

Subject: We got the contract!

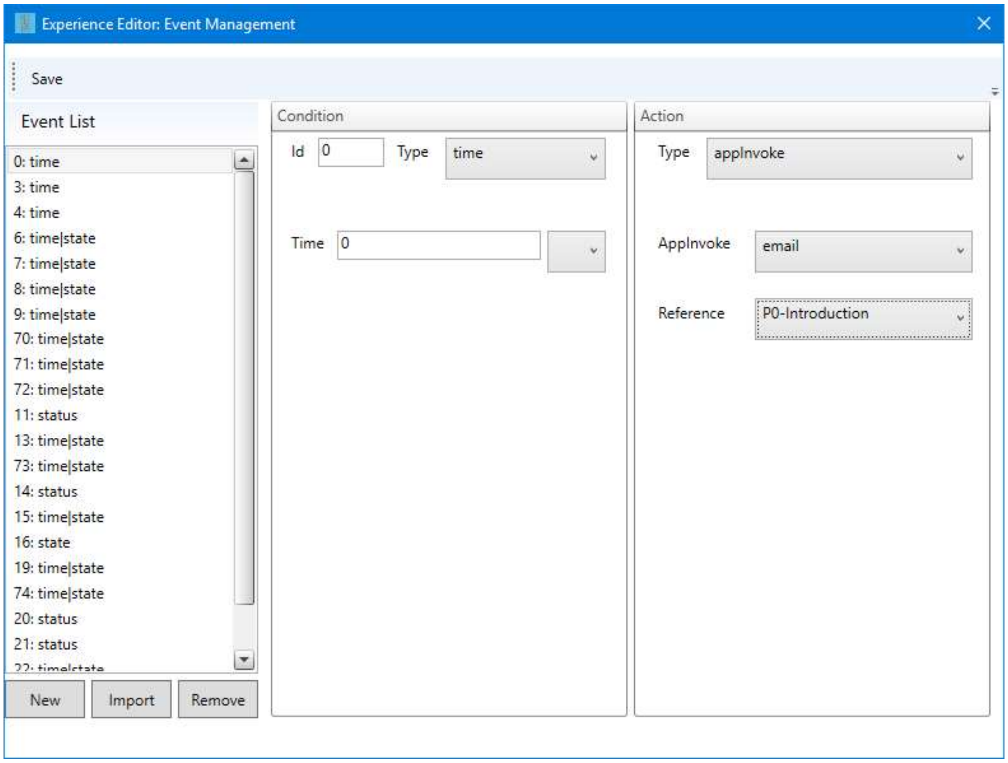
Body:

Dear Orville,

We just received a telegram from the Army Signal Corp. We got the contract! We have 5 month to make changes to the flying machine we have. Time to work!

Yours,
Wilbur

Figure 25 Mail Creation using Artifact Management Tool



Event List

- 0: time
- 3: time
- 4: time
- 6: time|state
- 7: time|state
- 8: time|state
- 9: time|state
- 70: time|state
- 71: time|state
- 72: time|state
- 11: status
- 13: time|state
- 73: time|state
- 14: status
- 15: time|state
- 16: state
- 19: time|state
- 74: time|state
- 20: status
- 21: status
- 22: time|state

Condition

Id: 0 Type: time

Time: 0

Action

Type: appInvoke

AppInvoke: email

Reference: P0-Introduction

Figure 26 Event Editor Tool for Event Creation

NOVICE EVALUATION OF EXPERIENCE BUILDING TOOLS VIA WRIGHT BROTHERS EXPERIENCE

The following are the evaluation results from the Wright Brothers Experience.

1. **Evaluation goals and objectives**
   The goal of this evaluation is to evaluate the use of the Experience Building Tool.
   The objective of this evaluation is to evaluate the instructions and ease of use to develop a simulated experience demonstrating decision impacts.

2. **Initial product assessments**
   The product did appear easy to use with the one-on-one guidance from an expert user. Once the novice user was on their own the development was difficult. A significant amount of time is required to obtain enough experience to properly use the Experience Editor tool. Specific detailed instructions are required to provide the opportunity to develop experiences without expert user assistance.

3. **Additional products or currently deployed solutions within the sponsor's environment worth considering**
   N/A

4. **Considerations/requirements for the sponsor's environment**
   The experience developer responsibilities require additional consideration. The current DAU ENG301 course has a significant amount of faculty responsibility in executing a course offering and it is unreasonable to expect faculty members to obtain to invest the time and labor necessary to become an expert user.

   We may consider developing a specific SEEA team at DAU to manage and develop experiences. It is unreasonable to expect a single point of contact to maintain these responsibilities.

5. **Evaluation criteria and the test plan**
   Step 1: Expert user provided one-on-one time using share screen with the novice user. This guidance was easy to follow. Basic examples were explored and found to be informative.

   Step 2: The novice user was provided a descriptive document instructing the user of how to use the Experience Building Tool for the Wright Brothers example experience.

   Step 3: The novice user attempted to build phases and subphases in the Events Management and had the following observations:
   1. Opening/Adding a New Phase and Subphase was clear and consisted of a basic drag and drop action.

2. The information required to build the phase is Name, Time, Contacts, Events, and Cycles.
    a. The Time has the option for Abs or Rel. The user needs additional description on how this impacts the experience.
    b. The Events has the option to Add, Remove, or Refresh. When choosing Add, a list of Events appears. It is unclear if these are default events. The user needs additional description on where these originate or if the user is required to develop a set to choose from.
    c. The Cycle has a field to enter the number of cycles. The user needs additional information on what a cycle consists of and how this field is used.
3. The Import Phase and Import Subphase options appeared to be options to import an existing element already developed and stored on a computer to be chosen by designating a specific file for import.

Step 4: The novice user attempted to build Artifacts in the Artifact Management and had the following observations:

1. Building a new Phase was unclear. It appears that the user can choose a Phase File, but it is unclear how these files are created. In addition the remaining fields require additional descriptions.
2. Building new Emails and Voicemails is unclear. Editing existing Emails and Voicemails appears to be clear and easy to edit.
3. The PDF files can be imported by selecting a file name for import. Additional information is needed to describe the Import SWF option.
4. The NPC option needs additional information to instruct the user to fill out each field and then click Add to save the new NPC.
5. Additional information is required to identify the origination of the CDR Criteria and the possibility of modifying the existing criteria.

FUTURE WORK

The evaluation of the tools has identified a number of areas for future work. There are two potential research thrusts moving forward. The first is in addressing issues that were identified in the evaluations, but were not addressed due to time constraints. The second is to use and update the tools in the development of a complete, new experience, thus providing a much more comprehensive evaluation.

FEATURES AND REFINEMENTS FOR SIMULATION MODELER

Future research for the simulation tools includes the following items:

- Sim Builder
    - Provide additions to sub-model library based on new experiences

    - Address usability and navigation issues identified in the evaluation but not yet addressed
    - Add story documentation to sub-models

- Sim Tuner
    - Include a recommendation form-like construct so that learner decisions can be tested more easily
    - Address usability and navigation issues identified in the evaluation but not yet addressed

- Chart Designer
    - Add color palettes for charts
    - Address usability and navigation issues identified in the evaluation but not yet addressed

- Additional testing and evaluation
    - Additional DAU personnel
    - Academic partners (GTRI, NPS, UAH)
    - Industry partners
    - System Dynamics Society

- Transition tools for use
    - Deploy via open-source license
    - Encourage continued use by academic and industry partners
    - Identify potential new users

FEATURES AND REFINEMENTS FOR EXPERIENCE BUILDING TOOLS AND LEARNING ASSESSOR

Future research for the Experience Building Tools and Learning Assessor includes the following items:

- Experience Building Tools
    - Add capabilities to create project status variable modification
    - Add functions to support user interface design
    - Add functionality to remotely integrate changes to server

- Learning Assessment Tool
    - Develop a performance assessment engine which evaluates learners' competency by comparing their performance to an experts' one.
    - Add function to generate an objective score based on the experience performance and decision making process.
    - Add function to measure the efficacy of the experience by comparing a learner's performance data with historic data.
    - Add function to visualize the decision making process of learners.

- Additional testing and evaluation by the following:
  - Stevens Institute of Technology
  - Additional DAU personnel
  - Academic partners (GTRI, NPS, UAH)
  - Industry partners
  - System Engineering Society

- Transition tools for use
  - Deploy via open-source license
  - Encourage continued use by academic and industry partners
  - Identify potential new users

NEW EXPERIENCE DEVELOPMENT

For the first category, it would be illuminating to use the tools to create a full experience based on the Wright Brothers assignment. The simulation model has demonstrated how this can be useful for the learner in terms of understanding a decision input and its effect on the outcome KPPs and TPMs. Similar models can be developed to address the schedule, cost and risk aspects of the Wright Brothers experience. In addition, the experience building tools can be used to design the phases, cycles and events that occur, while incorporating the various artifacts such as the contractual requirements, specifications of the original flyer, and templates for schedule and budget. Non-player characters can be used to provide feedback, hints and information above and beyond the handouts from the paper assignment. Finally, the learning assessor can be used to measure how well the learners have absorbed various concepts and skills.

CONCLUSION

This report has presented results from the third part of a project to create tools to support design, development and measurement of learning of experiences in the Experience Accelerator. Each Part has built on the successful completion of its previous part. The tools development efforts fall into four major categories:

- simulation tools for building and testing simulation models that mimic the behavior and results of acquisition programs that focus on system design and development,

- experience building tools that provide the structure for such system engineering experiences and the events that occur in them,

- learning assessment tools to measure the efficacy of the experience

- EA infrastructure changes to support the tools and evolution of the EA (HTM5 upgrade).

Table 1 shows the work plan for the three parts of the effort.

Table 1. SEEA tool development plan

Time	Simulation Tools	Experience Builder Tools	Learning Analysis Tools	EA Infrastructure
Part 1	Develop prototype Sim Builder and Tuner tools	Develop prototype Phase and Event Editor, and/or Artifact Integrator tools	Research and create high level design of Learning Assessor tool	Update EA infrastructure to support tools
Part 2	Extended functionality of Sim Builder and Tuner tools, make available to broader community. Determine new functionality as needed	Refine Phase and Event Editor, and/or Artifact Integrator tools, make available to broader community. Open source. Determine new functionality as needed	Develop prototype Learning Assessor prototype	Update EA infrastructure to support tools
Part 3	Extend functionality of Sim Builder and Tuner tools, make available open source. Determine new functionality as needed. Support existing tools, Specify new tools	Extend functionality of tools, make available open source. Support existing tools, Specify new related tools	Develop Learning Assessor tool, make available to for external evaluation. Specify future functionality	Update EA infrastructure to support tools

The risk and mitigation plan worked well in supporting the successful completion of all the objectives of this task that did not have dependencies on other work. In addition, a new capability not envisioned at the beginning of this project, the NPC Editor, was developed and evaluated. The report concludes with recommendations for future work. In particular, the focus on agile development was important in allowing desired and high-priority functionalities to be developed.

- **Risk**: Ensuring that the HTML5 conversion can be completed within the time window for Part 3 and the available funding.
- **Mitigation strategy**: This work will be prioritized such that risk is sufficiently reduced before beginning other Experience Builder and Learning Assessor tasks.
- **Outcome**: This strategy was successful, the major development HTML5 conversion was completed before the beginning of Experience Builder and Learning Assessor tasks.

- **Risk**: Lack of available potential tool users for tool evaluation, in particular simulation tool validation.
- **Mitigation strategy**: Use existing developers to evaluate the tools. Line up evaluators from DAU and other sources early in the project. For the simulation tools, validation could be done within the Increment 4 simulation work (e.g., reliability KPP, or post-CDR phase updates). Longer term, validation can be enhanced by releasing these tools to the general Systems Dynamics community, perhaps starting with the INCOSE working group.
- **Outcome**: This strategy was not needed, as the team was able to find a number of active users to conduct evaluation.

- **Risk**: Successful integration of the EA tools into the existing EA application infrastructure.
- **Mitigation strategy**: Continue to support an iterative approach in the integration of the current set of tools. Have someone who has a detailed understanding of the code and extensive experience in its operation implement the integration efforts.
- **Outcome**: This strategy was successful, as the tools code integrates into the EA infrastructure. This efforts was aided by the HTML5 conversion of the infrastructure being performed at the same time as tool development. In addition, the simulation tool code-based overlaps significantly with the simulation code in the EA infrastructure, so the simulation tool work addressed both the tools and the infrastructure.

- **Risk**: Ensuring the availability of graduate student staff with the knowledge and capabilities required for effective, efficient tool development despite gaps in funding.
- **Mitigation strategy**: Keep current graduate students engaged as much as possible with stop gap funding from other sources. Determine other sources of students or software research developers.
- **Outcome**: This strategy was successfully used prior to project start so that key graduate students were retained.

- **Risk**: The likelihood of feature creep that prevents the completion of an adequate set of tools for to provide a complete environment for Experience creation.
- **Mitigation strategy**: Utilize agile development practices to ensure that the highest value features are being developed at all times.
- **Outcome**: This strategy was successfully used.

The use and evaluation of the tools has been a very important part of the overall project effort. Much of the tool's definition, prioritization and subsequent development has been driven by the needs of actual development efforts including the existing DAU UAV experience, and the creation and development of new experiences namely the UK MoD Tempest, Project Robot and Systems Thinking Evaluation experiences. The use of the tools in each of these experiences has validated the value, and capabilities of the tools. In the case of the UK MoD Tempest and Project Robot experiences, the tools were used successfully people outside the Experience Accelerator team who had no prior experience with the tools. In addition, software programming skills were not necessary for any of this development work.

However, the tools were found to be difficult to use without additional support by a DAU evaluator who is representative of the DAU instructor population. In this case, it is likely that new experiences and the update of existing experiences should be handled by dedicated staff who have acquired experience in this area. However, certain features of the Experience Accelerator can be made accessible to instructors for tuning for their course. This could include the mode in how the experience is to be deployed (e.g., with individuals vs. teams, with or without instructor participation; the various difficulty levels, the number of iterations, how performance is to be valued, and the ways in which learner performance is being evaluated, etc.).

Future work is recommended in improving the tools based on the evaluation results through the continued development of the Wright Brother's experience. This process would entail the determination and creation of the training tools necessary to educate new developers and maintainers of the EA experiences.

APPENDIX A: LIST OF RESEARCH RELATED PUBLICATIONS

Bodner, D., Wade, J., Squires, A., Reilly, R., Dominick, P., Kamberov, G., & Watson, W. (2012). Simulation-based decision support for systems engineering experience acceleration. In *Proc. 2012 IEEE Systems Conference*, Vancouver, BC, Canada.

Bodner, D., Wade, J., Watson, B., Kamberov, G. (2013). Designing an experiential learning environment for systems engineering and logistics. In *Proc. 2013 Conference on Systems Engineering Research (CSER)*, Atlanta, GA.

Hao, K. (2015). Design and development tools for the experience accelerator. Thesis, Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ.

Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011a). Building a competency taxonomy to guide experience acceleration of lead program systems engineers. In *Proc. 2011 Conference on Systems Engineering Research (CSER)*, Redondo Beach, CA.

Squires, A., Wade, J., Watson, B., Bodner, D., Okutsu, M., Ingold, D., Reilly, R., Dominick, P., Gelosh, D. (2011b). Investigating an innovative approach for developing systems engineering curriculum: The Systems Engineering Experience Accelerator. In *Proc. 2011 American Society for Engineering Education (ASEE) Annual Conference and Exposition*, Vancouver, BC.

Squires, A., Wade, J., Watson, W., Bodner, D., Reilly, R., Dominick, P. (2012). Year one of the Systems Engineering Experience Accelerator. In *Proc. 2012 Conference on Systems Engineering Research (CSER)*, Rolla, MO.

Turner, R., Bodner, D., Kemp, D., Rodriguez, Y., Wade, J., Zhang, P. (2017). SE Simulation Experience Design: Infrastructure, Process and Application. In *Proc. 27th Annual INCOSE International Symposium*, Adelaide, Australia.

Wade, J. P., Bodner, D. A., Turner, R., Hinkel, J., & Zhang, P. (2016). RT 146: Design and Development Tools for the Systems Engineering Experience Accelerator – Part 2. Technical Report No. SERC 2016-TR-108, Systems Engineering Research Center, Hoboken, NJ.

Wade, J., Kamberov, G., Bodner, D., Squires, A. (2012). The architecture of the Systems Engineering Experience Accelerator. In *Proc. International Council on Systems Engineering (INCOSE) 2012 International Symposium/European Conference on Systems Engineering (EUSEC)*, Rome, Italy.

Zhang, P., Bodner, D. A., Turner, R. G., Arnold, R. D., Wade, J. P. (2016). The Experience Accelerator: Tools for development and learning assessment. In *Proc. of the 2016 American Society for Engineering Education (ASEE) Annual Conference and Exposition*. Washington, DC: ASEE.

Zhang, P., Wade, J., Turner, R., Bodner, D., Thomas, D. (2017). SEEA: Accelerated Learning and Learning Assessment for Systems Engineering Education. In *Proc. 2017 Conference on Systems Engineering Research*, Los Angeles, CA.

APPENDIX B: CITED AND RELATED REFERENCES

Journal and Conference Articles and Reports

Ardis, M., McGrath, E., Lowes, S., & Lam, S. (2012). Research on building education & workforce capacity in systems engineering. Interim Technical Report SERC-2012-TR-019-1, Systems Engineering Research Center, Stevens Institute of Technology, Hoboken, NJ.

Armstrong, J., & Wade, J. (2015). Learning systems engineering by teaching it. In *Proc. 2015 INCOSE International Symposium*, 211-220.

Arnold, R., & Wade, J. (2015). A definition of systems thinking: A systems approach. *Procedia Computer Science*, 44, 669-678.

Baker, A., Navarro, E. O., & van der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, 75(1-2), 3-16.

Belland, B. R., French, B. F., & Ertmer, P. A. (2009). Validity and problem-based learning research: A review of instruments used to assess intended learning outcomes. *Interdisciplinary Journal of Problem-Based Learning*, 3(1). Available at: http://dx.doi.org/10.7771/1541-5015.1059.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.

Burandt, S. (2011). Effects of an educational scenario exercise on participants' competencies of systemic thinking. *Journal of Social Sciences*, 7, 54-65.

Chapman, G. M., Martin, J. F. (1995). Computerized business games in engineering education. *Computers & Education*, 25(1-2), 67-73.

Chen, W.-F., Wang, T.-L., Su, C.-H., & Wu, W.-H. (2008). Work in progress - A game-based learning system for software engineering education. In *Proc. 2008 IEEE Frontiers in Education Conference*.

Ebner, M., & Holzinger, A. (2007). Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education*, 49(3), 873-890.

Futter, M. (2014). [Correction] World of Warcraft subscriptions slide by 800,000. Game Informer. GameStop. Retrieved August 5, 2014.

Hopper, M. & Stave, K. A. (2008). Assessing the effectiveness of systems thinking interventions in the classroom. In *Proc. 26th International Conference of the System Dynamics Society*.

Jain, A., & Boehm, B. (2006). SimVBSE: Developing a game for value-based software engineering. In *Proc. 2006 Conference on Software Engineering Education & Training*.

Klieme, E., & Maichle, U. (1991). Erprobung eines Modellbildungssystems im Unterricht. Bericht über eine Pilotstudie zur Unterrichtsevaluation. Bonn: Institut für Test- und Begabungsforschung.

Kopainsky, B., Alessi, S. M., Pedercini, M., & Davidsen, P. I. (2009). Exploratory strategies for simulation-based learning about national development. In *Proc. 27th International Conference of the System Dynamics Society*.

Kopainsky, B., Pirnay-Dummer, P., & Alessi, S.M. (2012). Automated assessment of learners' understanding in complex dynamic systems. *System Dynamics Review*, 28(2), 131-156.

Kopainsky, B., & Saldarriaga, M. (2012). Assessing understanding and learning about dynamic systems. In *Proc. 30th International Conference of the System Dynamics Society*.

Kopainsky, B., & Sawicka, A. (2011). Simulator-supported descriptions of complex dynamic problems: Experimental results on task performance and system understanding. *System Dynamics Review*, 27, 142-172.

Morsi, R., & Jackson, E. (2007). Playing and learning? Educational gaming for engineering education. In *Proc. 2007 IEEE Frontiers in Education Conference*.

Okutsu, M. (2009). Teaching engineering design principles via a serious game format. Report, Purdue University, West Lafayette, IN.

Ossimitz, G. (2000). Teaching systems dynamics and systems thinking in Austria and Germany. *Proc. 18th International Conference of the System Dynamics Society*.

Pirnay-Dummer, P., Ifenthaler, D., & Spector, J. M. (2010). Highly integrated model assessment technology and tools. *Educational Technology Research and Development*, 58(1), 3-18.

Plate, R. (2010). Assessing individuals' understanding of nonlinear causal structures in complex systems. *System Dynamics Review*, 26, 19-33.

Rimiene, V. (2002). Assessing and developing students' critical thinking. *Psychology Learning & Teaching*, 2, 17-22.

Rouwette E.A.J.A., Größler A., Vennix J.A.M. (2004). Exploring influencing factors on rationality: a literature review of dynamic decision-making studies in system dynamics. *Systems Research and Behavioral Science*, 21(4), 351–370.

Skaza, H., & Stave, K. (2009). A test of the relative effectiveness of using systems simulations to increase student understanding of environmental issues. In *Proc. 27th International Conference of the System Dynamics Society*.

Skaza, H., & Stave, K. A. (2010). Assessing the effect of systems simulations on systems understanding in undergraduate environmental science courses. In *Proc. 28th International Conference of the System Dynamics Society*.

Spector, J. M. (2006). A methodology for assessing learning in complex and ill-structured task domains. *Innovations in Education and Teaching International*, 43(2), 109-120.

Spector, J. M., Christensen, D. L., Sioutine, A. V., & McCormack, D. (2001). Models and simulations for learning in complex domains: Using causal loop diagrams for assessment and evaluation. *Computers in Human Behavior*, 17(5), 517-545.

Stave, K. A. (2011). Using simulations for discovery learning about environmental accumulations. In *Proc. 29th International Conference of the System Dynamics Society*.

Stave, K. A., Beck, A., & Galvan, C. (2014). Improving learners' understanding of environmental accumulations through simulation. *Simulation & Gaming*, 46(3-4), 270-292.

Steif, P. S. H. M. (2006). Comparisons between performances in a statics concept Inventory and course examinations. *International Journal of Engineering Education*, 22(5), 1070-1076.

Sterman, J. D. (2010). Does formal system dynamics training improve people's understanding of accumulation? *System Dynamics Review*, 26, 316-334.

Stewart, M. (2012). Joined up thinking? Evaluating the use of concept-mapping to develop complex system learning. *Assessment & Evaluation in Higher Education*, 37(3), 349-368.

Taran, G. (2007). Using games in software engineering education to teach risk management. In *Proc. 2007 Conference on Software Engineering Education & Training*.

Walbridge, M. (2008). Analysis: Defense of the Ancients – An Underground Revolution. http://www.gamasutra.com/, retrieved June 23, 2008.

Weinerth, K., Koenig, V., Brunner, M., & Martin, R. (2014). Concept maps: A useful and usable tool for computer-based knowledge assessment? A literature review with a focus on usability. *Computers & Education*, 78, 201-209.

Books and Book Chapters

Cohen, S. (2006). *Virtual Decisions: Digital Simulations for Teaching Reasoning in the Social Sciences and Humanities*. Mahwah, NJ: L. Erlbaum Associates.

Driscoll, M. P. (2005). *Psychology of Learning for Instruction* (3rd Ed.). Boston: Pearson Education, Inc.

Dörner, D., & Wearing, A. (1995). Complex problem solving: Toward a (computersimulated) theory. In *Problem Solving: The European Perspective*, Frensch, P. A., & Fubke, J. (eds.), Psychology Press, 65-102.

Gibson, D., Aldrich, C., & Prensky, M. (eds.) (2007). *Games and Simulations in Online Learning: Research and Development Frameworks*. Hershey, PA: Information Science Pub.

Herrington, J., & Oliver, R. (1995). Critical characteristics of situated learning: Implications for the instructional design of multimedia. In *Learning with Technology*, Pearce, J., & Ellis, A. (Eds.). Parkville, Vic: University of Melbourne, 235-262.

Herz, J. C., & Macedonia, M. R. (2002). Computer games and the military: Two views. *Defense Horizons*, 11.

Jones, K. (1997). *Games and Simulations Made Easy: Practical Tips to Improve Learning through Gaming*. London: Kogan Page.

Karakul, M., & Qudrat-Ullah, H. (2008). How to improve dynamic decision making? Practice and promise. In *Complex Decision Making: Theory and Practice*, Qudrat-Ullah, H., Spector J.M., & Davidsen, P.I. (eds). Springer/NECSI: Berlin; 3–24.

Kolb, D. (1984). *Experiential Learning*. Prentice Hall, Inc.

Novak, J. D., & Gowin, D. B. (1984) *Learning How to Learn*. Cambridge University Press, Cambridge.

O'Neil, H. F., & Perez, R. S. (2008). *Computer Games and Team and Individual Learning*. Amsterdam: Elsevier.

Prensky, M. (2001). *Digital Game-based Learning*. New York: McGraw-Hill.

Shaffer, D. W. (2006). *How Computer Games Help Children Learn*. New York: Palgrave Macmillan.

Van Ments, M. (1999). *The Effective Use of Role-Play: Practical Techniques for Improving Learning*. London: Kogan Page.

APPENDIX C: SIMULATION MODELER EVALUATION FEEDBACK

The detailed comments and recommendations for the Simulation Modeler are shown below. They are divided into three categories – under interface, model editing functions, and new features/functions. An update is provided for those items that have been addressed.

User Interface

1. System Description – I see there is a way to "Add description to submodel" through a toolbar button, but I do not know how to actually view that description. It would be nice to have a description associated with the system; just a textual narrative describing, in general, what this system is supposed to represent.

   Update – Users may view the description using the "add description" function, which also enables them to edit the description.

2. Mouse Wheel Zoom – I would like to be able to zoom in and out on the models using mouse wheel (not just the zoom buttons). I have my wide screen monitor vertically oriented so I do not have a lot of horizontal space. The models do not fit very well. Also, I like to zoom in and out frequently as I look at things so I prefer to be able to use an easily accessible control rather than clicking a button.

3. Sub-Model Context Menu (Right-Click) – Right-clicking on empty space within a sub-model should bring up a context menu that provides access to some useful options; a few suggestions:
   - Close Sub-Model
   - New-> (all the node types)
   - Enter "Add Flow" Mode

4. Additions to Node Context Menu – I'd like to see some additions to the context menus of the nodes, including:
   - Close Sub-Model
   - New-> (all the node types)
   - Enter "Add Flow" Mode

5. Node Parameter Editor (Double-Click on Node) – It would be nice to be able to double-click on a node to bring up a full menu of that node's properties (name, start, min/max, curve type, etc.; everything it has in one editor screen). Sometimes I am finding it a little difficult to visualize the nodes holistically without being able to see all of the information together. The hover-over capability helps but it does not stay on screen long enough.

6. Timing of Mouse Hover Over Node – I do like seeing the information on the node upon mouse hover, but I'd recommend changing the timings – the information should appear more quickly (personally I would prefer instant), and should stay visible until the mouse

is moved away or clicked (similar to the way mouse hover works in most modern browsers and programs).

7. Click a Node in Change Formula (Rate Node) – After clicking "Change formula" on a Rate Node, the formula screen appears. On this screen, I would like to be able to just click on a node in the list on the right rather than having to type its ID into the formula text field.

8. Resizing a Node (Visual) – I wish I could change the sizes of the nodes (by clicking and dragging their edges) just for visual reasons; a large portion of my node names get cut off by their size constraints.

9. Close Sub-Models Separately – I wish I could close sub-models one at a time instead of all at once. I have accidentally created an extra sub-model several times (suggestion for changing button position of "create new sub-model" button below) and to "get it off the screen" I seem to have to close all the sub-models at once using the File->Close option.

10. Move the "Create New Submodel" Tool Bar Button – The "Create New Submodel" button in the tool bar is in an inconvenient place; I would put it all the way to the right the edit tools, between "Change model name" and "Execute" and in its own space, separated by a vertical separator. I pressed it several times by accident.

    Update – This has been addressed via changing the order of the toolbar buttons.

11. File->New – It is not immediately clear what "new thing" the File->New option in the file menu actually creates. I would recommend including a sub-menu here of all the different new things that can be created; or at least, change "New" to "New Sub-Model" to make it more clear.

12. File->Close – I recommend changing the Close option in the file menu to "Close All" and add an option for "Close Selected Sub-Model"

13. Resize Sub-Model Viewing Area – I wish I was able to resize the viewing area of the sub-models. Right now it seems that each imported or open sub-model takes up equal space within the Sim Builder viewing area. I would like to be able to click and drag the viewing area of one model to make it larger while the other becomes smaller, just like the way most programs with multiple viewing windows operate (Visual Studio, Eclipse, File Explorer, etc.).

    Update – This has been addressed by allowing the user to double-click on the sub-model being edited. This sub-model is then brought up in a new window for editing.

14. Infinite Max Value on Level Nodes – I would like the ability to remove the max values on level nodes. Although I agree that technically all variables in the universe (probably) have a max value, practically speaking, level node maxes sometimes can be so high as to be effectively infinite for the purposes of the system under design. Either an "infinite" or a "no max" option would be fine. Otherwise I'm finding myself specifying arbitrarily

large values (9999999) just to keep the max high enough that it's never reached during my simulation.

Update – To let the maximum (minimum) value be infinity (negative infinity), the user can simply leave the field blank.

15. Change NULL to "Normal" for Curve Type – I recommend changing the "NULL" option on the Curve Type (within a level node) to just "Normal" and I would put "Normal" at the top of the list of 3.

16. Saving Last File Path - Default Model Location in File System – It would be nice if the Sim Builder program remembered the location of the last model I opened, and defaulted to that folder when I clicked open or import. Navigating back to the folder that contains my models every time I click Open or Import is a huge pain, especially when it defaults to some esoteric folder like "Documents." The information could be saved as part of the Options file, wherever the language preference is stored.

    Update – This has been implemented.

17. Node Name Display – I'd like an option to be able to display the entire name text for the nodes instead of truncating the name. Although it might look "messy" when the name text extends out of the bounds of the node drawing area, I still think it'd be easier to see than having it all truncated.

    Update – The user may view the full-name by hovering the tooltip over the node icon. Many variable names are quite long in complex models.

18. Formula Tool Tips (Auxiliary and Rate Nodes) – In the "Enter new formula" text field in an auxiliary or rate node, when a user hovers over the text field, it would be nice if a tooltip showing an example of a formula would appear. For example, "This sample formula does XYZ: <formula>" just to make it a little more user friendly for new users. Along these lines, perhaps a "help" button near the text field that provides a detailed description of how to write the various different formulas, what formulae are available, etc.

<u>Model Editing Functions</u>

19. Delete Key – It would be nice to be able to use the delete key as an alternate to the "Remove" option. It should delete anything selected, including groups of things.

20. Deleting / Removing Items – Sometimes when I try to remove things, the system blocks the action due to various dependencies. I would prefer the system to recurse over the various dependencies and remove everything dependent (perhaps with a confirmation message – "Flows XYZ will be removed, accept / reject?"

21. Chart Designer Plan Node and Plan Node Increment – I would recommend changing "Length" under Plan Node Increment in the Chart Designer to "Length (# of Rounds)" or just Number of Rounds. Also, I'd right-justify the text with their corresponding edit boxes instead of left-justifying them (Name, Id, Start Value, etc.).

    Update – The text has been changed.

22. Only One Selection at Once Across All Models – I've noticed it's possible to select a node on one model, then select another node on another model at the same time. The problem is, the use of the various tool bar buttons then becomes ambiguous – when I click "Share" for example, which node am I sharing? I would recommend only allowing one selection at one time across all models (like the way it works now across a single model, but globally).

<u>New Features/Functions</u>

23. Validate Button – In my opinion, the Sim Builder should probably not try to validate the system models when saving or loading; it should validate on Execute, and there should be a "Validate" button that allows the user to attempt to validate the model on demand. During the design and production of a model, there may be many times in which an invalid model is present. That is going to be a natural part of the construction process. I found myself wanting to save my work, but unable to do so because my current version of the model was apparently invalid (although strangely, sometimes I was able to save an invalid model – but then it would not load). As a user, I just want to be able to hit "Save" and have all my current stuff save, then come back later, hit "Load," and have it all there – valid or not.

24. Visually Sub-Dividing the System into Manageable Parts – I am not sure if there's already a way to do this that I'm missing, but I think this suggestion is going to be important in the future. However, it may take a fair amount of design work to get it right.
    - The issue is that I feel like there needs to be some way to sub-divide large systems into parts, yet still maintain their holism. I thought shared nodes would let me do this. But, it isn't possible to create a flow coming from a shared node. I can kind of see why this makes sense from a technical standpoint (you'd sort of have to "pull in" the entire model that the node is shared from into the current model in order to affect its value), but, this can make the implementation of even relatively simple systems quite complex.
    - Visually speaking there just needs to be a way to simplify sets of interactions. For example, I wish I could take a group of nodes that results in a single level node output and just collect them all as, for example, "Food Gatherer." On screen, I could choose to simply show one "Node Grouping" called "Food Gatherer" and it would appear as, maybe, a circle. With this technique, I could get a good idea of how my system worked holistically but also drill down into the various pieces. Ideally this could be done ad infinitum, so you could end up with a very complex system that, on screen, only looked like three circles. But it

would still be possible to expand a circle to see a whole sub-system system inside. If the entire set of components in a group is self-contained (with all flows and connections resulting in the single level node), I think this should be straightforward. If not, then the flows can simply be drawn from an entire group to wherever they should go (node or group, depending on how the user has chosen to organize the visual layout).

- Basically, any lines and nodes internal to the group just will not draw. Anything that flows to/from the group to the outside, or feeds an outside thing, will just draw from the highest-level grouping. I think this type of hierarchical visualization capability will be really important in the design of new systems using this tool, especially more complex ones.
- Another option is to allow an import of an entire other sub model to just one node - but not as a “shared node” per se, because it will run as part of the current model (flowing to/from the level node as part of the simulation).